



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV INTELIGENTNÍCH SYSTÉMŮ**

DEPARTMENT OF INTELLIGENT SYSTEMS

**MODELOVÁNÍ NAD GENEALOGICKÝMI DATY**

MODELLING FOR GENEALOGY

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. HANA PROSTREDNÍKOVÁ**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**doc. Ing. FRANTIŠEK ZBOŘIL, Ph.D.**

**BRNO 2018**

**Vysoké učení technické v Brně - Fakulta informačních technologií**

Ústav inteligentních systémů

Akademický rok 2017/2018

**Zadání diplomové práce**

Řešitel: **Prostředníková Hana, Bc.**

Obor: Informační systémy

Téma: **Modelování na základě genealogických dat  
Modelling for Genealogy**

Kategorie: Modelování a simulace

**Pokyny:**

1. Seznamte se se entitami, vztahy a rolemi, které se objevují v zápisech matričních a jiných historických knih využívaných v genealogii.
2. Navrhněte strukturu databáze, tak, aby byla vhodná pro uchovávání dat získaných z výše uvedených knih.
3. Nalezněte pravidla, která musí být platná pro rodokmenové struktury (např. že rodič musí být starší než potomek) a ve vhodně zvoleném programovacím jazyce vytvořte systém, který umožní kontrolovat platnost těchto pravidel v daném rodokmenu.
4. Rozšiřte systém tak, aby uživatel mohl pokud možno co nejsnadněji formulovat vlastní dotazy nad rodokmenem.
5. Experimentujte s dodanými daty, ověřte funkčnost a použitelnost vytvořeného systému a diskutujte dosažené výsledky.
6. Vytvořte plakát velikosti A1, na kterém shrnete výsledky vaší práce, zejména fungování vytvořeného systému.

**Literatura:**

- Lednická, B.: Sestavte si rodokmen, pátráme po svých předcích, Grada, 2012
- Leon, S., S.: The Art of Prolog, MIT Press, 1994

Při obhajobě semestrální části projektu je požadováno:

- Studium současných systémů, návrh

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Zbořil František, doc. Ing., Ph.D., UITS FIT VUT**

Datum zadání: 1. listopadu 2017

Datum odevzdání: 23. května 2018

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav inteligentních systémů  
612 66 Brno, Božetěchova 2

doc. Dr. Ing. Petr Hanáček  
vedoucí ústavu

## Abstrakt

Tato práce obsahuje podrobnou studii dané problematiky týkající se genealogie jako vědy a genealogických záznamů. V práci jsou rozebrány role a vztahy vyskytující se v genealogických záznamech a je popsána problematika jejich reprezentace. Cílem je návrh a implementace systému, který ověří platnost vztahů v genealogických záznamech a umožní další práci s těmito daty.

## Abstract

This thesis contains detailed study of given problems related to genealogy science and genealogical records. There are analyzed roles and relationships that occurs in genealogical records and problems of their representation are described too. The goal is to design and implement system, which will validate relationships in genealogical records and enable processing this data.

## Klíčová slova

Genealogie, modelování, Prolog, dotazování

## Keywords

Genealogy, modelling, Prolog, query

## Citace

PROSTREDNÍKOVÁ, Hana. *Modelování nad genealogickými daty*. Brno, 2018. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce doc. Ing. František Zbořil, Ph.D.

# Modelování nad genealogickými daty

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracovala samostatně pod vedením pana doc. Ing. Františka Zbořila, Ph.D. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

.....  
Hana Prostředníková  
21. května 2018

## Poděkování

Děkuji vedoucímu práce doc. Ing. Františku Zbořilovi, Ph.D., rodině a příteli za podporu a pořádnou dávku optimizmu.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Studium problematiky</b>	<b>4</b>
2.1	Cíl práce . . . . .	4
2.2	Pojem genealogie a její význam v historii . . . . .	4
2.3	Základní pojmy . . . . .	6
2.4	Typy genealogických záznamů . . . . .	9
2.5	Role a vztahy mezi nimi . . . . .	11
2.6	Vztahy mezi osobami z formálního hlediska . . . . .	13
2.7	Problémy reprezentace genealogických dat . . . . .	13
2.8	Práce s matričními záznamy . . . . .	15
2.9	Jazyk Prolog . . . . .	16
<b>3</b>	<b>Analýza požadavků a návrh systému</b>	<b>20</b>
3.1	Popis uživatele a přehled očekávaných funkcí . . . . .	20
3.2	Prvotní analýza požadavků . . . . .	21
3.3	Use case diagram a systémový sekvenční diagram . . . . .	21
3.4	Funkční požadavky . . . . .	21
3.5	Další požadavky . . . . .	21
3.6	Specifikace struktury dat . . . . .	23
3.7	Pravidla platná ve vztazích . . . . .	26
<b>4</b>	<b>Implementace výsledného systému</b>	<b>27</b>
4.1	Struktura programu . . . . .	27
4.2	Vložení vstupních dat . . . . .	28
4.3	Ověření konzistence dat . . . . .	29
4.4	Statistiky . . . . .	31
4.5	Rekonstrukce genealogických struktur . . . . .	33
4.6	Vyhledání vztahu mezi dvěma osobami . . . . .	37
4.7	Export databáze do souboru . . . . .	39
4.8	Manuál . . . . .	40
4.9	Další vývoj . . . . .	40
<b>5</b>	<b>Experimenty nad výsledným systémem</b>	<b>42</b>
5.1	Ověření konzistence dat . . . . .	42
5.2	Vyhledání otcovské/mateřské linie . . . . .	44
5.3	Určení vztahu mezi dvěma osobami . . . . .	45
5.4	Statistiky . . . . .	47

<b>6 Závěr</b>	<b>49</b>
<b>Literatura</b>	<b>50</b>
<b>Přílohy</b>	<b>52</b>
<b>A Obsah přiloženého CD</b>	<b>53</b>
<b>B Use-case diagram</b>	<b>54</b>
<b>C Systémový sekvenční diagram</b>	<b>55</b>

# Kapitola 1

## Úvod

V současnosti stoupá zájem společnosti o genealogii. Děje se tak především díky zveřejňování genealogických knih na internetu. Informace jsou tak mnohem dostupnější veřejnosti, než tomu bylo v minulosti. Není potřeba navštěvovat archivy osobně a celý výzkum se tak stává finančně i časově méně náročný. Genealogické knihy obsahují velké množství dat, které je problematické studovat. Nabízí se proto myšlenka automatického zpracování takových dat.

Cílem této práce je návrh systému, který bude definovat role osob v genealogických záznamech a bude vymezovat vztahy mezi nimi. Vztahy budou popsány řadou pravidel, která je vymezují. Pomocí těchto pravidel bude možné ověřit platnost genealogických struktur. Také bude možno automaticky generovat genealogické struktury z matričních záznamů.

Práce je členěna do čtyř částí. V první je definován cíl práce a uvedena je také studie problematiky. Je definován pojem genealogie a popsán její význam v průběhu dějin. Dále jsou definovány základní pojmy, jejichž znalost je potřebná pro další práci. Uvedeny jsou jednotlivé typy genealogických knih a je rozebrána problematika reprezentace genealogických dat. Vysvětleny jsou také pojmy vztah a role v genealogických strukturách a tyto pojmy jsou popsány i z formálního hlediska. V závěru kapitoly je popsán programovací jazyk Prolog a je rozebrána vhodnost jeho použití při implementaci systému.

Druhá část se zabývá analýzou požadavků a návrhem výsledného systému. Nejprve je uveden typický profil předpokládaného uživatele. Provedena je komplexní analýza požadavků. Nejprve jsou analyzovány základní požadavky, které plynou z cíle práce. Dále jsou analyzovány očekávané funkce systému a na jejich základě jsou uvedeny funkční požadavky. Důležitou částí této kapitoly je přesná specifikace vstupních dat. Na závěr jsou uvedena pravidla platná ve vztazích, která bude systém dodržovat a ověřovat jejich platnost.

Implementace výsledného systému je popsána ve čtvrté kapitole. Nejprve je uvedeno členění kódu do jednotlivých souborů. Dále jsou popsány možnosti importování genealogických databází. Následuje popis implementace kontroly konzistence vstupních dat. Vysvětlen je výpočet statistik v jednotlivých databázích. Dále jsou rozebrány jednotlivé algoritmy pro rekonstrukci genealogických struktur a pro vyhledání vztahu mezi dvěma osobami. Je popsán způsob změny dat v databázi a možnosti jejich exportu do souboru. V závěru kapitoly je vysvětlen způsob zobrazení uživatelské nápovědy a jsou vytyčeny cíle dalšího vývoje.

Poslední kapitola obsahuje výsledky experimentů s výsledným systémem ve všech implementovaných oblastech. V závěru této práce jsou shrnuty dosažené cíle.

## Kapitola 2

# Studium problematiky

Obsahem této kapitoly je definice cíle práce a studium dané problematiky. Je vysvětlen pojem genealogie a píše se o úloze, kterou tato věda sehrávala v průběhu dějin. Dále jsou uvedeny základní pojmy, které budou používány v práci. Následuje přehled způsobů, kterými jsou genealogická data zaznamenávána a jsou uvedeny názorné příklady. Popsány jsou také problémy spojené s reprezentací genealogických dat a způsob práce s nimi. V závěru kapitoly jsou popsány principy jazyka Prolog.

### 2.1 Cíl práce

Cílem práce bude vytvoření systému, který bude rozlišovat role a vztahy v záznamech z matričních knih. Systém bude definovat omezení, platná pro jednotlivé vztahy, a tyto vztahy bude následně ověřovat. Systém bude implementován v programovacím jazyce Prolog (kap. 2.9), který je pro tento typ úlohy velmi vhodný. Umožňuje definovat omezení platná pro jednotlivé vztahy formou pravidel. Dle těchto pravidel budou následně vyhodnocovány dané dotazy. Možné bude také vytváření uživatelských dotazů kombinací existujících pravidel.

Hlavní myšlenkou této práce je definice vztahů mezi osobami v různých rolích a následné vyhledávání genealogických struktur v genealogických datech. Systém bude předpokládat data v konkrétní podobě (popsána dále). Správná funkčnost celého systému bude záviset na kvalitě dodaných dat.

### 2.2 Pojem genealogie a její význam v historii

Malá československá encyklopedie [15] definuje pojem genealogie jako jednu z pomocných věd historických, která studuje rodové a rodinné souvislosti jednotlivců a vztahy hospodářské, právní a kulturní povahy, jež z těchto vztahů vyplývají. Samotné slovo genealogie pochází z řeckého slova *génos*, jenž znamená rod.

Znalost vlastních předků byla, je a vždy bude důležitou informací o nás samotných. Již v historii sehrávala genealogie důležitou roli. V průběhu celých dějin lidstva se po celém světě setkáváme s genealogickými záznamy. Výborným příkladem toho, jak důležité byly v minulosti rodinné vazby a jakou roli sehrávalo, z jakého rodu kdo pochází, je samotná Svátá Bible [16]. Již ve Starém zákoně se setkáváme s podrobným popisem Adamových synů a jejich potomků. Studium se lze dopátrat až k Noemovi, který je přímým potomkem Adama. Dále pak v knize Numeri jsou uvedeny jednotlivé rody spolu se spisy o jejich majetku. Jsou zde popsány také zákony, které přikazují ženám, provdat se za muže patřícího do jejich



rodu, aby jejich majetek nepřešel do rodu jiného. Z toho je patrné, že již v takto vzdálené historii genealogie hrála významnou roli, kromě jiného, v rámci hospodářských vztahů. S tímto jevem se můžeme setkat v průběhu celé historie. Jedná se zejména o dohodnuté sňatky z finančních důvodů. Stejně tak se popisy rodových linií objevují i v Novém zákoně. Evangelium dle Matouše podrobně popisuje rodokmen od Abraháma po Ježíše. V evangeliu podle Lukáše je zase naopak popsána linie od Ježíše až po samotného Adama. Podobné detailní popisy lze najít i v jiných knihách popisujících světová náboženství.

Genealogie však je a byla až znepokojivě často využita jako nástroj moci a k šíření nenávistných myšlenek. Jedním z nejhorších příkladů zneužití genealogických informací v našich zemích je bezpochyby nacistická ideologie. V roce 1935 byli v německém Norimberku přijaty Říšským sněmem dva zákony, které se do historie zapsali jako tzv. norimberské zákony [2]. První z těchto zákonů stanovil, že občanem Říše může být pouze Němec, nebo člověk s příbuznou krví. Druhý zákon potom zakázal, kromě jiného, sňatky mezi Němci a Židy. Vyhlášením těchto zákonů byl každý občan povinen dokázat svůj nežidovský původ při státní službě, sňatku či jiných právních úkonech (obr. 2.1). Dle norimberských zákonů byla rozhodující příslušnost prarodičů k židovské náboženské obci. Každý, kdo pocházel z alespoň tří židovských prarodičů, byl automaticky považován za tzv. plného Žida. Dále pak vznikla kategorizace míšenců a položidů. Takové osoby měli obdobná práva jako němečtí občané, avšak vztahovala se na nich celá řada omezení (např. možnost sňatku pouze s jiným položidem). Jednalo se tedy o cílenou diskriminaci na základě rodového původu.

Obrázek 2.1: Příklad dokumentu k potvrzení nežidovského původu. [3]

## 2.3 Základní pojmy

V této kapitole se budeme věnovat základním pojmům genealogie, jejichž pochopení je nezbytné pro další studium tohoto dokumentu. Informace uvedené v této kapitole jsou přebrány ze zdroje [12].

### Probant (střen)

Probant, nebo také střen, je osoba, od které začíná genealogický výzkum. Nejčastěji je tento člověk v genealogických schématech označován jako „já“ (lat. ego). Probantem ale může být také osoba stojící na počátku genealogického schématu (rodokmenu/rozrodu, vysvětleno dále), tedy nejvzdálenější předek, od kterého je odvozován původ ostatních jedinců.

### Ascendenti

Jedná se o souhrnné označení osob, které jsou pokrevně spřízněné ve vzestupné linii. Tedy linie přímých předků probanta.

### Descendenti

V tomto případě hovoříme o potomcích v přímé sestupné linii od probanta.

### Příbuzenství dle krve a práva

Příbuzenství lze definovat jako vztah mezi osobami, který je založený na pokrevních vazbách a z nich plynoucích dalších právních vztazích. Pokrevním příbuzenstvím (lat. consanguinitas) rozumíme vztah, kdy jedna osoba pochází přímo od druhé. Rozlišujeme přímou linii (otec, syn), a boční linii, která je založená na společném předkovi (sourozenci, bratřenci, apod.)

Do příbuzenstva se však počítají i osoby, ke kterým nemáme žádné pokrevní vazby, ale existují jiné dvě osoby, které v pokrevním vztahu jsou. Hovoříme o tzv. právním příbuzenství. V genealogii je pro takový vztah vžitý termín „švagrovství“. Mezi osoby v tomto vztahu se řadí např. tchán, tchýně, švagr, zeť, nevěsta, svat, atd. Je potřeba rozlišovat také zákonné příbuzenství, které patří do okruhu právního příbuzenstva. Tento pojem je znám také jako adopce, tedy osvojení si dítěte.

### Ztráta předků

Zde hovoříme o jakémsi fenoménu, ke kterému dojde každý badatel v průběhu svého výzkumu, pokud hledá dostatečně dlouho. Platí, že každý z nás má dva rodiče, čtyři prarodiče, atd. Pokud se na to podíváme z matematického hlediska, získáme rovnici č. 2.1.

$$A_n = 2^n \tag{2.1}$$

V této rovnici  $n$  představuje číslo generace, přičemž probant je nultá generace. Nyní by tedy mělo stačit postupně vypočítat počet předků pro každou generaci a sečtením těchto hodnot bychom měli získat reálný počet všech našich předků. Zde však nastává problém. Tabulka 2.1 ukazuje, kolik předků by v dané generaci i celkově mělo být. Výpočet v tabulce předpokládá průměr 25 let na jednu generaci.

Číslo generace	Přibližné roky	Předkové v této generaci	Celkový počet předků
1	25	2	2
2	50	4	6
3	75	8	14
4	100	16	30
38	950	274 877 906 944	549 755 816 886
39	975	549 755 813 888	1 099 511 627 774
40	1000	1 099 511 627 776	2 199 023 255 550

Tabulka 2.1: Přehled předpokládaného počtu předků v různých generacích

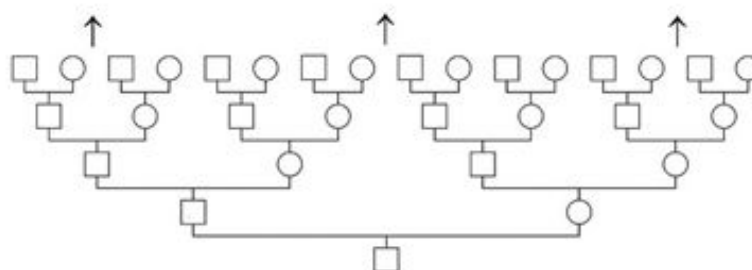
Dle tabulky tedy po tisíci letech dostaneme číslo o něco větší než dva biliony předků. Toto číslo má zásadní problém, značně převyšuje počet všech lidí, kteří kdy na planetě Zemi žili. Zde se tedy dostáváme k pojmu *inbreeding*. Jedná se o uzavírání manželství mezi geneticky příbuznými partnery. Takto dochází ke zmíněné ztrátě předků, které říkáme *implex*. Při úbytku předků se některé osoby objevují v různých generacích. Implex sledujeme častěji v populacích s nízkým okruhem potenciálních partnerů. Taktéž se velmi často vyskytuje v šlechtických rodech, kde hrály roli politické a mocenské zájmy. Proto velmi často docházelo ke sňatkům např. mezi přímými bratřenci a sestřenicemi.

## Generace

V genealogii považujeme za generaci všechny osoby ve stejné vzdálenosti od probanta. Někdy tímto pojmem rozumíme také pokrevně svázané osoby žijící v přibližně stejném časovém období.

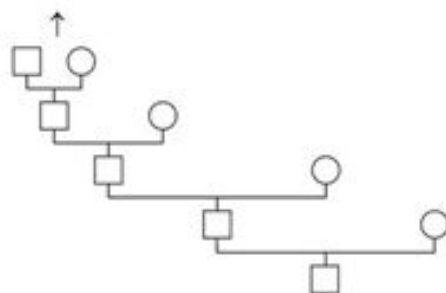
## Rodový vývod (vývod z předků)

Jedná se o soupis všech přímých předků probanta a to jak v mužské tak v ženské linii. Zaznamenává všechny manželské páry, které se podílely na zplození sledované osoby (obr. 2.2).



Obrázek 2.2: Schéma vývodu z předků. [17]

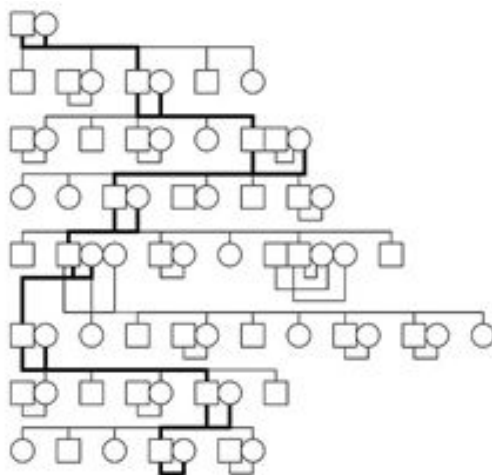
Pokud budeme brát v úvahu pouze otcovskou linii (obr. 2.3), hovoříme o *agnantním vývodu* (otec, děda, praděda...). Obdobně pak pokud budeme sepisovat pouze mateřskou linii (matka, babička, prababička...), hovoříme o *kognantním vývodu*.



Obrázek 2.3: Schéma agnantního vývodu. [17]

## Rodokmen

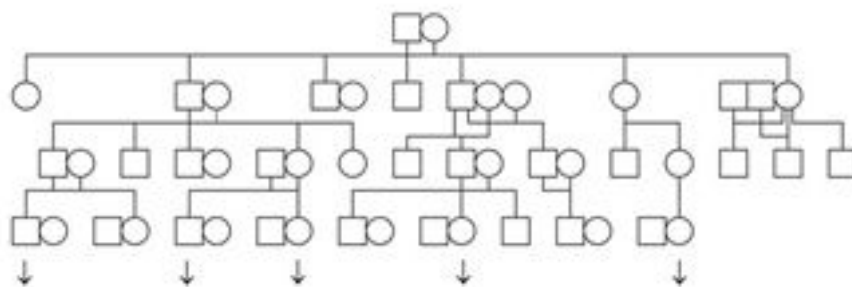
Jedná se o soupis všech nositelů stejného příjmení. Do rodokmenu se tedy zahrnují manželské páry a jejich potomci, přičemž se sleduje otcovská linie od probanta až po nejstaršího dohledatelného předka. Znamená to tedy, že rodokmen obsahuje potomky pouze toho manželského páru, ve kterém figuruje přímý mužský předek probanta nesoucí jeho příjmení (obr. 2.4). V současnosti jsou pojmem rodokmen chybně označovány všechny druhy genealogických tabulek.



Obrázek 2.4: Schéma rodokmenu. [17]

## Rozrod

Rozrod představuje nejkomplexnější soupis potomků pocházejících od jednoho konkrétního předka, případně manželského páru (obr. 2.5). Soupis je veden směrem do současnosti a zahrnuje všechny mužské i ženské potomky. Není proto výjimkou, že obsahuje velmi velký počet osob a různá příjmení. Rozrod je zajímavý také z hlediska genetického či demografického (dědičné choroby, dlouhověkost, volba povolání, migrace, apod.).



Obrázek 2.5: Schéma rozrodu. [17]

## Genetická genealogie

Pod tímto pojmem rozumíme využití poznatků z genetiky při genealogickém výzkumu. Tento vědní obor lze chápat jako sledování stupně genetické příbuznosti sledovaných osob, geografického původu předků, případně lze sledovat migraci určitých genetických skupin.

Výzkum se provádí analýzou DNA (deoxyribonukleová kyselina), která se nachází v jádru somatických buněk. Rozborem DNA lze zjistit, jaký je stupeň příbuzenství mezi dvěma sledovanými osobami (např. potvrzení otcovství). Dále pak lze zjistit, ze kterých míst na zemi pocházeli předkové sledované osoby.

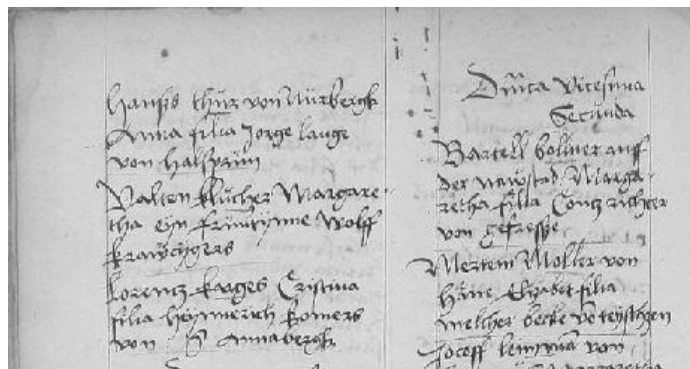
Vzorkem DNA jsou typicky stěry z ústní dutiny, lze však využít i jiné vzorky (krev, sliny, kůže, kořen vlasu, ...). Poté, co je ze vzorku DNA extrahována, dojde k jejímu sekvencování, tedy zjištění sekvence nukleových bází, ze kterých se DNA skládá a následné dekodování informace. Při analýze DNA lze sledovat otcovskou linii (chromozom Y), anebo mateřskou linii (mitochondriální DNA).

## 2.4 Typy genealogických záznamů

Hlavním zdrojem při pátrání po předcích jsou matriky. Jedná se o záznamy o narozeních, sňatcích a úmrtích osob v daném městě či obci. V současnosti jsou tyto záznamy vedeny matričními úřady. První vedené matriky pocházejí již ze středověku. V té době však neexistovalo žádné nařízení k jejich vedení, a proto se objevovaly málo. Navíc se většina z nich nedochovala. V našich zemích platí, že lze jen velmi obtížně nalézt záznamy z období před bitvou na Bílé hoře (1620). Povinnost vést matriční záznamy uložil všem farářům Tridentský koncil, který se uskutečnil již roku 1563. Od koncilu k realitě však vedla dlouhá cesta. V našich zemích o jejich vedení rozhodla nejdříve synoda olomoucká v roce 1591 a později synoda pražská v roce 1605. Řádně však začaly být matriky vedeny až po ukončení třicetileté války, tedy přibližně od poloviny 17. století. [1]

Za nejstarší dochovanou matriku je považována kniha narozených města Jáchymov [4]. Nejstarší záznam v této knize pochází z roku 1531, tedy dlouho před zasedáním Tridentského koncilu. Město v té době prožívalo velký rozmach kvůli rozsáhlým nálezům stříbra. Informace o těžbě se rychle roznesly i do jiných zemí a vlivem migrace se z Jáchymova jen za pár let stalo po Praze druhé největší město. V samotné knize nacházíme kromě českých jmen také německá i jiná ze vzdálenějších koutů Evropy. Na obrázku (obr. 2.6) je zobrazena ukázka zápisu v této matrice. Informace uvedené v následujících podkapitolách byly nastudovány ze zdrojů [1] a [12].





Obrázek 2.6: Ukázka zápisu v nejstarší dochované matriční knize [4]

## Kniha narození

V této matriční knize byly zapisovány narozené děti (obr. 2.7). V zápisu se uváděl datum narození, jméno dítěte, jméno a příjmení otce a jméno matky. Dále zde pak byly informace o křtu, tedy kdy a kde byl uskutečněn, jméno a příjmení křtícího a jména kmotrů a kmoter. Hlavním kmotrem (kmotrou) byl v matričním zápisu označován ten, kdo dítě držel při samotném křtu. V záznamu je tato osoba označena jako *compater* (*conmater*) *levans*. Ostatní kmotři a kmotry byli označováni jako *testes* (svědkové).

1847				1848				1849				1850			
Geb.				Geb.				Geb.				Geb.			
Datum				Datum				Datum				Datum			
Zeit				Zeit				Zeit				Zeit			
Ort				Ort				Ort				Ort			
Vater				Vater				Vater				Vater			
Mutter				Mutter				Mutter				Mutter			
Name				Name				Name				Name			
Stand				Stand				Stand				Stand			
1. August 1847				1. August 1848				1. August 1849				1. August 1850			
Hans von Hinz				Hans von Hinz				Hans von Hinz				Hans von Hinz			
Anna von Hinz				Anna von Hinz				Anna von Hinz				Anna von Hinz			
1. August 1847				1. August 1848				1. August 1849				1. August 1850			
Hans von Hinz				Hans von Hinz				Hans von Hinz				Hans von Hinz			
Anna von Hinz				Anna von Hinz				Anna von Hinz				Anna von Hinz			

Obrázek 2.7: Ukázka záznamu z knihy narozených obce Libice nad Cidlinou z roku 1847[10]

V některých případech (často na Slovensku) byla vedena kniha křtů samostatně. Pokud neexistoval záznam o narození, bral se jako datum narození datum křtu. V minulosti totiž byly děti kvůli vysoké úmrtnosti křtěné hned v den narození. Matka se tohoto obřadu zpravidla nezúčastňovala. Kromě toho, že byla několik dní po porodu zesláblá a odkázaná na lůžko, byla také v období šestinedělí považována za „nečistou“ a její přítomnost v chrámu nebyla možná.

## Kniha manželství

Kniha manželství (obr. 2.8), jak název napovídá, zaznamenávala důležité informace o sňatcích. Tedy datum a místo sňatku, jména a příjmení snoubenců, rodičů a oddávajícího, vyznání a stav (svobodný, vdovec, ...) snoubenců. Dále zde byly uvedeny jména svědků, kteří byli při obřadu přítomni. Nakonec zde bylo zaznamenáno, jak a kdy byly provedeny ohlášky, případně důvod, proč byli novomanželé od ohlášek osvobozeni.

Rok 1845		Místo		Jméno		Příčina smrti		Poznámky	
číslo	čas	číslo	čas	číslo	čas	číslo	čas	číslo	čas
1	29. 8.	1	29. 8.	1	29. 8.	1	29. 8.	1	29. 8.
Záznamy z knihy manželství obce Libice nad Cidlinou z roku 1845									

Obrázek 2.8: Ukázka záznamu z knihy manželství obce Libice nad Cidlinou z roku 1845<sup>[10]</sup>

## Kniha zemřelých

Do této knihy byly zapisovány úmrtí osob (obr. 2.9). Záznamy o zemřelých byly v minulosti velmi strohé, často pouze datum úmrtí a jméno zemřelého. Později byly přidávány další informace, jako místo kde se zemřelý narodil a žil, nebo jméno manžela/manželky. Zajímavou informací je také uvedení příčiny smrti. Vzhledem k tomu, že zápis prováděl většinou farář, přesnost této informace závisela na jeho znalostech. Často se proto u novorozenců vyskytuje jako příčina smrti „vrozená slabost“ a u starých lidí jednoduše „přirozená smrt“.

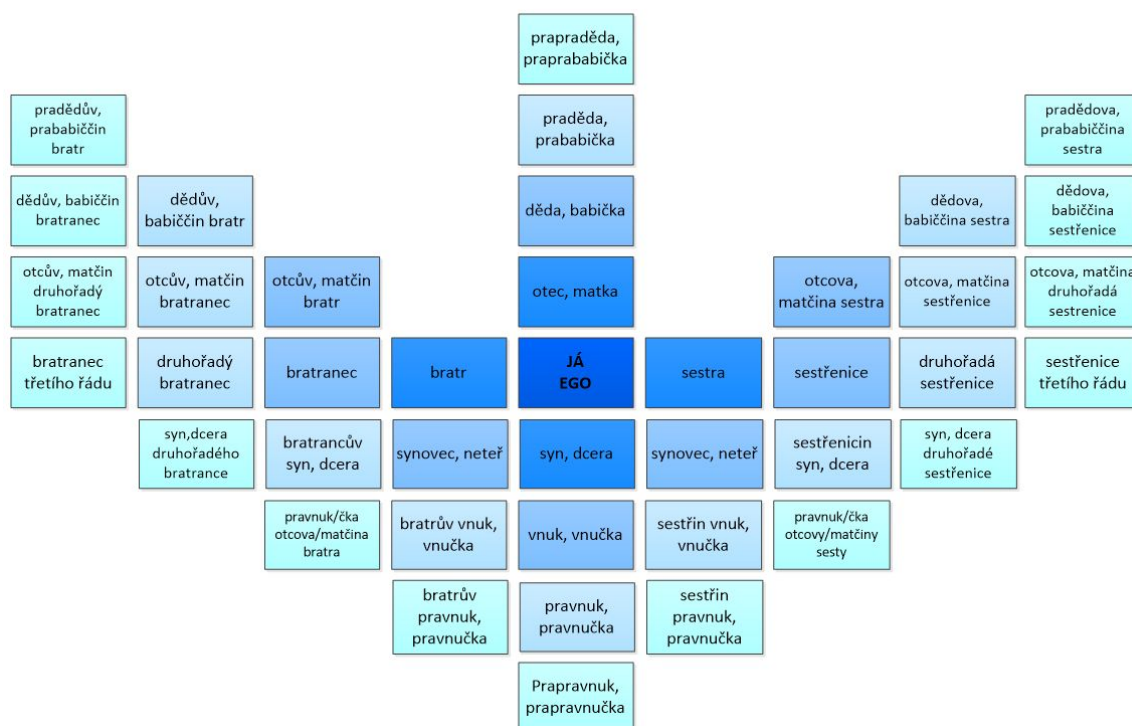
Rok 1844		Místo		Jméno		Příčina smrti		Poznámky	
číslo	čas	číslo	čas	číslo	čas	číslo	čas	číslo	čas
1	29. 8.	1	29. 8.	1	29. 8.	1	29. 8.	1	29. 8.
Záznamy z knihy zemřelých obce Libice nad Cidlinou z roku 1844									

Obrázek 2.9: Ukázka záznamu z knihy zemřelých obce Libice nad Cidlinou z roku 1844<sup>[10]</sup>

## 2.5 Role a vztahy mezi nimi

Při tvorbě rodokmenu provádíme vlastně soupis našich předků, resp. potomků a zaznamenáváme vazby mezi nimi. Velmi rychle si uvědomíme, že každá osoba vystupuje v mnoha

různých rolích. Proto je osoba ega v genealogickém výzkumu důležitá a musí být jasné, kdo jí je. Obrázek 2.10 zobrazuje schéma příbuzenských vztahů blízké rodiny. Je opravdu důležité, z kterého pohledu se na osobu díváme. Například o mé matce já tvrdím, že je to má matka, ale moje sestřenice bude tvrdit, že je to její teta (tedy sestra jejího otce či matky). Moje matka tedy pro mne a moji sestřenici sice představuje jednu a tu samou osobu, jenže v jiné roli. Na druhé straně já i má sestřenice budeme souhlasit, že moje babička je i její babička. To proto, že je to náš společný předek. Můj potomek by však už s tímto tvrzením samozřejmě nesouhlasil. Čím rozsáhlejší rodokmen je, tím větší je počet rolí, do kterých každá osoba vstupuje. Nyní jsme samozřejmě uváděli pouze pokrevní příbuzenstvo. Pokud bychom brali v úvahu také právní příbuzenstvo (tedy nevěsty, zetě, ...) dostaneme rolí mnohem více.



Obrázek 2.10: Schéma příbuzenských vztahů dle zdroje [12]

Na základě těchto rolí odvozujeme vztahy mezi dvěma osobami. Pojem role a vztah se v tomto slova smyslu značně překrývají. Vztahy mezi osobami se řídí určitými zákonitostmi, které jsou pro každý vztah jiné. Můžeme zcela jistě říct, že ve vztahu rodič-potomek bude rodič určitě starší než jeho potomek o nějaký minimální počet let (předpokládejme 15). Toto omezení ale nemusí platit pro vztah strýc/teta-synovec/neteř, kde se klidně může stát, že obě osoby budou například stejně staré. Jiným příkladem může být, že dva pokrevní sourozenci (tedy mající stejné rodiče) musí být narozeni minimálně 9 měsíců po sobě (bereme-li v úvahu extrémní případy). Toto ale neplatí pro dva bratrance/sestřenice. Mezi různými rolemi tedy platí jiná pravidla, která jsou pro daný vztah specifická. Pokud bychom se na vztahy podívali z právního hlediska i zde platí přesná pravidla. Například vyživovací povinnost ve vztahu rodič-potomek, zákaz manželství mezi sourozenci apod.



## 2.6 Vztahy mezi osobami z formálního hlediska

Pokud bychom chtěli genealogické role a vztahy nějakým způsobem formalizovat, je vhodné uvést pojem relace. Následující výklad byl nastudován ze zdroje [14].

**Definice 2.6.1** *N-ární relace je libovolná podmnožina kartézského součinu  $n$  množin.*

**Definice 2.6.2** *Relace na množině  $A$  je libovolná množina  $R$  uspořádaných  $n$ -tic, přičemž platí  $R \subseteq A^n$ .*

Dle uvedené definice 2.6.1 tedy platí, že binární relace je libovolná podmnožina kartézského součinu dvou množin. Binární relace na množině  $A$  je potom dle 2.6.2 libovolná podmnožina kartézského součinu  $A \times A$ . Binární relace  $R$  na množině  $A$  může mít řadu vlastností. Uvedme si některé, které nás budou zajímat. Říkáme, že relace  $R$  je:

- **reflexivní** -  $\forall a \in A : (a, a) \in R$
- **symetrická** -  $\forall a, b \in A : (a, b) \in R \wedge (b, a) \in R$
- **tranzitivní** -  $\forall a, b, c \in A : (a, b) \in R \wedge (b, c) \in R \Rightarrow (a, c) \in R$

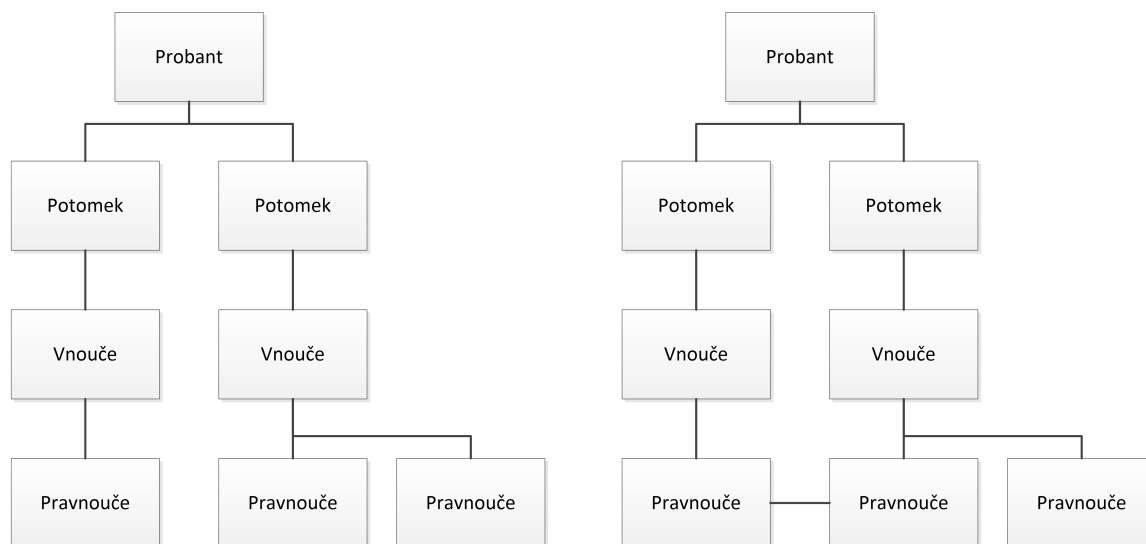
Relace na množině, která má výše uvedené vlastnosti se nazývá *relace ekvivalence*. Mějme nyní množinu  $M = \{m_1, m_2, \dots, m_n\}$  osob. Dále mějme relaci  $R_m$ , která je relací na množině  $M$  a určuje pokrevní příbuzenstvo osob z množiny  $M$ . Podívejme se například na prvky  $m_1, m_2$  a  $m_3$ . Řekněme, že osoba  $m_1$  má za otce osobu  $m_2$ , a osoba  $m_2$  má za otce osobu  $m_3$ . S jistotou můžeme říct, že pokud platí, že  $(m_1, m_2) \in R_m$  potom platí také  $(m_2, m_1) \in R_m$ . Jinými slovy, pokud je osoba  $m_1$  potomkem osoby  $m_2$  potom platí že osoba  $m_2$  je otcem osoby  $m_1$ . Obdobně platí i  $(m_2, m_3) \in R_m \wedge (m_3, m_2) \in R_m$ . Relace  $R_m$  je tedy **symetrická**. Dle uvedeného jistě platí  $(m_1, m_2) \in R_m \wedge (m_2, m_3) \in R_m \Rightarrow (m_1, m_3) \in R_m$ , tedy že osoba  $m_1$  je přes svého otce pokrevně spřízněna s osobou  $m_3$ , která je jejím prarotcem. Můžeme tedy říci, že relace  $R_m$  je **tranzitivní**. Nakonec je každá osoba pokrevně spřízněna sama se sebou, takže je splněna i poslední vlastnost, tedy **reflexivita**. Relace  $R_m$  je tedy relace ekvivalence.

## 2.7 Problémy reprezentace genealogických dat

V minulé kapitole jsme si uvedli různé druhy matričních knih. Všechny mají společný rys, a to, že záznamy jsou za sebou řazeny tak jak přicházely. Na tom není nic špatného, je problematické zapisovat záznamy do matriky tak, aby vytvářely nějakou komplexní strukturu. Ta je právě cílem genealogického výzkumu. Při pátrání je úkolem genealoga vyhledat v matrikách relevantní informace. To znamená vyhledat správnou osobu, která odpovídá hledané roli a vztahu vůči výchozí osobě. Jedná se o zdlouhavý a náročný proces, kterého výsledkem má být nějaké genealogické schéma (uvedeno v kap. 2.2). Jak ale lze toto schéma vhodně reprezentovat? Pokud je výsledkem genealogického výzkumu výstup čítající poměrně malý počet osob, je v lidských silách zakreslit dané schéma rukou, případně použitím nějakého grafického editoru. Jedná se o poměrně jednoduchou úlohu, vzhledem k malému počtu nalezených vztahů. V případě rozsáhlejšího výzkumu zde však nastává problém. V ideálním světě by měl být výstupem výzkumu strom (def. 2.7.1).

**Definice 2.7.1** *Obyčejný souvislý graf, jehož žádný podgraf není kružnicí, se nazývá **strom** [18].*

Pokud se však stane, že v rodině došlo k příbuzenskému sňatku, v rodokmenu se vytvoří cyklus a výše uvedená definice přestává platit. Takovéto sňatky nebyly v minulosti ničím výjimečným a i v současnosti se s nimi můžeme setkat. Obrázek 2.11 ukazuje, jak se situace změní příbuzenským sňatkem. V pravé části došlo k sňatku mezi dvěma právnoúčaty probanta. Vztah mezi nimi je sestřenice a bratranec druhého řádu. Sňatkem změnili své role na manžele. Čím vyšší je počet nalezených osob a čím více do minulosti se při výzkumu dostaneme, tím více takových sňatků nalezneme.



Obrázek 2.11: Příklad stromové struktury vlevo a příklad porušení této struktury vpravo.

To znamená, že dostáváme rozsáhlou grafovou strukturu, která již není stromem ale obecným grafem, ve kterém je velmi obtížné se vyznat. V takové chvíli je vhodné sáhnout po nějakém genealogickém programu, který výrazně usnadní práci s velkým objemem dat. Takových programů existuje celá řada. Jejich vzhled či funkcionality se do jisté míry liší, ale v současnosti má většina těchto programů společnou vlastnost: export dat do souboru formátu GEDCOM.

## GEDCOM

GEDCOM je zkratkou pro Genealogical Data Communication. Jedná se o detailní specifikaci formátu souboru pro ukládání a přenos genealogických dat. Specifikace byla vytvořena Církví Ježíše Krista Svatých posledních dnů (mormoni). Tato církev má již řadu let velký zájem na genealogickém výzkumu a digitalizaci matrik. I když od roku 2002 nevyšla žádná nová verze specifikace GEDCOM, stále je nejpoužívanějším formátem pro ukládání a přenos genealogických dat. [8]

GEDCOM je čistě textový soubor, který je označen příponou .ged. Takový soubor obsahuje informace o osobách a popisuje vztahy mezi nimi. Soubor začíná tzv. hlavičkou. Ta obsahuje informace ohledně programu, který soubor vygeneroval, jeho verzi, verzi GEDCOM specifikace apod. Po hlavičce následují jednotlivé záznamy. Tyto záznamy se dělí na záznamy o osobách a záznamy o rodinách. V záznamu osoby se může vyskytovat velké množství informací o dané osobě (jméno, příjmení, datum narození/úmrť, zaměstnání, místo narození, ...). Dále se zde vyskytují křížové odkazy, které říkají, do které rodiny osoba patří. V záznamu o rodině jsou pak přesně popsány vztahy mezi členy rodiny. V záznamu

o rodině může figurovat pouze jeden otec a jedna matka (tedy manželský pár) a neomezený počet potomků. Pokud například došlo v rodině k rozvodu a manžel si založil jinou rodinu, bude pro tuto rodinu založen nový záznam. Osoba manžela tedy bude figurovat ve dvou záznamech. Soubor je ukončen záznamem TRLR, který označuje konec všech záznamů a nemůže se v souboru vyskytovat víckrát.

Specifikace GEDCOM [8] je pro reprezentaci genealogických dat velmi vhodná. Jedná se o jednoduchou strukturu, která není složitá na zpracování. Jednotlivé záznamy jsou ukládány po řádcích (Obrázek 2.12). Na začátku každého řádku se nachází číslo, které určuje zanoření v rámci záznamu. Nový záznam tedy vždy začíná číslem 0, další informace v něm číslem 1. Pokud nějaká informace obsahuje další strukturovaná data, jsou uvedena na samostatných řádcích s číslem 2 atd. Specifikace je velmi komplexní a v současnosti ji skutečně podporuje většina genealogického software. V dokumentacích jsou uvedeny také specifikace GEDCOM X, což je upravený GEDCOM formát do formátu XML. Takový formát je ještě vhodnější pro používání vzhledem k velmi dobré čitelnosti a zpracovatelnosti XML souborů.

<u>Level</u>	<u>Cross-Ref. Identifier</u>	<u>Tag</u>	<u>Value</u>
0	@1@	INDI	
1		NAME	John Quentin/Doe/
1		FAMS	@4@
0	@2@	INDI	
1		NAME	Mary Ann/Wilson/
1		FAMS	@4@
0	@3@	INDI	
1		NAME	John/Doe/
1		FAMC	@4@
0	@4@	FAM	
1		HUSB	@1@
1		WIFE	@2@
1		CHIL	@3@

Obrázek 2.12: Ukázka struktury GEDCOM souboru[7]

## 2.8 Práce s matričními záznamy

Současné programy pro práci s genealogickými daty nabízejí uživateli širokou funkcionalitu. Genealog tedy při svém pátrání může nalezené osoby a vztahy jednoduše přímo zapisovat do takového programu. Ten pak nad nalezenými osobami dokáže automaticky vykreslovat různá schémata, např. vývod od konkrétní zvolené osoby apod. Tato funkcionalita je však podmíněna znalostí o vztahu mezi jednotlivými osobami. Podívejme se tedy na problém z jiné strany. Matriční knihy obsahují jednotlivé záznamy zapsané za sebou, tak jak chronologicky přicházely. Předpokládejme knihu narození nějakého konkrétního města, která obsahuje záznamy za nějaké větší časové období (dejme tomu 250 let). Za předpokladu, že v minulosti lidé velmi často zůstávali celý život v jednom městě, a že jedna generace trvala přibližně 25 let, měly by záznamy obsahovat přibližně 10 generací. Z genealogického hlediska by tedy bylo zajímavé automatické vyhledávání rodových linií přímo ze záznamů matrik.

Taková myšlenka má samozřejmě své problémy. Prvním, snad nejvýraznějším, je převod matriční knihy do elektronické podoby. Většina matričních knih byla psána ručně ještě v nedávné minulosti. Rukopis každého faráře či matrikáře se výrazně liší. Také styl psaného písma se v průběhu historie značně změnil. Knihy jsou navíc často poškozené, písmo rozmazané a špatně čitelné. Automatický převod nasnímaných záznamů do digitální podoby je tedy značně problematický.

Dalším problémem je pak nepřesnost záznamů. Velmi často kvůli selhání lidského faktoru. Špatně zapsané jméno, místo či datum narození nejsou ničím výjimečným. V datech tak nastávají nekonzistence, které jsou těžko odhalitelné.

Vyhledávat osoby dle jejich vzájemného vztahu je v takovýchto datech poměrně složitá úloha. V knihách narození je u narozeného dítěte sice napsáno jméno otce a matky, problémem je, že často chybí informace o jejich datu narození. V záznamu matky dokonce může scházet rodné příjmení. Vyhledání rodičů potom může být založeno pouze na předpokládaných vlastnostech jejich vztahu k dítěti. Tedy například, že rodiče musí být starší než dítě alespoň o přibližně 15 let, že musí být rodiče na živu v čase jeho narození (v případě otce lze tolerovat narození dítěte nejpozději do devíti měsíců po jeho smrti). Takových pravidel existuje celá řada. Jakým způsobem ale tato pravidla aplikovat a získat relevantní data? K tomuto účelu se nabízí programovací jazyk Prolog.

## 2.9 Jazyk Prolog

Prolog je programovací jazyk, který byl vytvořen v roce 1972 dvojicí autorů Alain Colmermauer a Robert Kowalski. Samotný název je zkratkou z „Programming in Logic“. Jazyk Prolog je tedy logický programovací jazyk. Řadíme ho do skupiny deklarativních programovacích jazyků. Prolog umožňuje uživatelům popsat úlohu logicky, místo psaní počítačových instrukcí. Programátor tedy nemusí říct, co všechno se má udělat pro to, aby byla úloha vyřešena, stačí, když řekne jak má řešení vypadat. Postup je pak ponechán na systému, který se k výsledku dostane odvozováním z faktů a pravidel v databázi [9].

Jazyk Prolog má poměrně jednoduchou syntaxi. Založen je na predikátové logice prvního řádu, přičemž se omezuje pouze na Hornovy klauzule. Jen pro úplnost je vhodné dodat, že tvar Hornových klauzul odpovídá následující definici.

**Definice 2.9.1** *Tvar Hornových klauzul je z pohledu predikátové logiky tento:*

$$H \leftarrow B_1 \wedge B_2 \wedge \dots \wedge B_n, n > 0 \quad (2.2)$$

kde  $H$  je hlavička a  $B_1 \wedge B_2 \wedge \dots \wedge B_n$  je tělo tvořené atomickými formulemi a  $H$  je faktem. [13]

Program zapsaný v jazyce Prolog je potom zapsán dle následujících pravidel:

- Klauzule:  $h \text{ :- } b_1, b_2, \dots, b_n$ .
- Fakta:  $h$ .
- Dotazy, cíle ve tvaru  $b_1, b_2, \dots, b_n$ .  
kde je levá implikace doplněna samostatně interpretem Prologu (výzvou ve tvaru ?-) [13]

Z uvedeného je patrná podobnost zápisu Hornovy klauzule v jazyce Prolog a jejího matematického zápisu. Levá implikace je reprezentována dvojicí znaků  $\text{:-}$  a logická konjunkce je reprezentovaná čárkou. Každá klauzule, fakt anebo cíl jsou zakončeny tečkou.

## Datové typy

Jazyk Prolog nepracuje s datovými typy tak, jako jiné jazyky. Rozlišit lze *atomy* (konstanty), *proměnné*, *termy* a *seznamy*. Atomem rozumíme celé číslo anebo posloupnost znaků začínající malým písmenem. Dále také posloupnost znaků uzavřenou v apostrofech.

Proměnné jsou tvořeny posloupností znaků a celých čísel. Proměnná musí začínat velkým písmenem anebo podtržítkem. Speciální funkci má potom proměnná, která je tvořena pouze podtržítkem. Tato reprezentuje tzv. anonymní proměnnou, která se používá, pokud není potřeba znát konkrétní hodnotu dané proměnné. Proměnná je v tomto jazyce vždy lokální. Rozlišujeme pojem *volná* a *vázaná proměnná*. Vázanou proměnnou rozumíme takovou proměnnou, která je vázaná na konkrétní hodnotu. Volná proměnná je taková, která nemá vazbu na hodnotu. Toto označení plyne z predikátové logiky, kde jsou obdobně definovány volné a vázané proměnné ve formuli.

V predikátové logice je term chápán následovně:

**Definice 2.9.2** *Term v predikátové logice je:*

- Každá proměnná
- Výraz  $f(t_1, t_2, \dots, t_n)$  je-li  $f$   $n$ -ární funkční symbol a  $t_1, t_2, \dots, t_n$  jsou termy
- Každý výraz získaný konečným počtem aplikací předchozích dvou pravidel, nic jiného term není [13]

V Prologu se term skládá z hlavičky a seznamu parametrů (tedy odpovídá druhému bodu definice 2.9.2). Počet parametrů termu nazýváme aritou termu, term je tedy identifikován hlavičkou a aritou.

Zajímavým typem v prologu je *seznam*. Jedná se o strukturovaný typ, kterého začátek i konec je označen hranatými závorkami. Jednotlivé prvky seznamu jsou odděleny čárkou. Prvkem seznamu mohou být atomy, či seznamy, přičemž seznamy mohou být libovolně zanořené. [13] [9]

## SDL rezoluce a unifikace

Na začátek vysvětlíme pojem *unifikace*. V Prologu je unifikace základní operací pro vyhodnocování. Díky ní se dostáváme k výsledku. Probíhá podle Robinsonova unifikačního algoritmu. Unifikaci používáme v následujících případech:

- Přiřazení  $\langle \text{hodnota} \rangle = \langle \text{proměnná} \rangle$
- Test na rovnost  $\langle \text{term1} \rangle = \langle \text{term2} \rangle$
- Selektor položek ze seznamů  $\langle \text{seznam} \rangle = [\langle \text{hlavička\_seznamu} \rangle | \langle \text{tělo\_seznamu} \rangle]$
- Vytváření seznamů  $\langle \text{proměnná} \rangle = [1, 2, 3]$
- Předávání parametrů hodnotou či odkazem
- Sdílení proměnných, vytváření synonym

Vyhodnocování dotazů se v Prologu provádí tzv. *SDL rezolucí*. Jednotlivé podcíle jsou expandovány do hloubky. Pokud je tedy hledán nějaký podcíl cíle, prohledávají se hlavičky klauzulí vložených do programu shora dolů, dokud není možné některou z hlaviček

unifikovat s podcílem. Pokud došlo k unifikaci, je tělo klauzule prováděno zleva doprava, přičemž první term na pravé straně je opět vyhodnocován do hloubky. Pokud jsme dospěli do situace, kdy jsou všechny atomy na pravé straně uspokojeny, hovoříme o *uspění podcíle* a následně dochází k uspění cíle. Stejně tak pokud je na pravé straně klauzule fakt. V případě, že nebyly všechny atomy na pravé straně těla klauzule uspokojeny anebo nelze unifikovat při prohledávání hlaviček klauzulí, nastává *selhání cíle*. V této situaci je pak vyvolán proces navracení (backtracking). Vyhodnocovací mechanismus se postupně vrátí do místa prohledávání a snaží se najít jinou hlavičku, kde by uspěl. Pokud se tímto způsobem nepodaří nalézt hlavičku, kde by došlo k unifikaci, selhává i hlavní cíl. [13] [9]

## SWI-Prolog

SWI-Prolog je všestranná implementace jazyka Prolog. Jedná se o multiplatformní interpret tohoto jazyka, který je velmi oblíbený především pro svou rychlost. Vestavěné predikáty splňují normu ISO. SWI-Prolog obsahuje komplexní rozhraní pro jazyk C. Program zapsaný v jazyce C může vyhodnocovat predikáty Prologu. Prolog tak může být využit jako vestavěný logický vyhodnocovací stroj. Toto rozhraní tvoří základ pro rozhraní jazyků vyšší úrovně (např. C++, Java, ...). SWI-Prolog dále obsahuje rozsáhlý webový server, který umožňuje vytváření webových aplikací založených na HTTP, JavaScript a CSS. Mezi další významné vlastnosti SWI-Prolog patří také vícevláknové zpracování, dobrá přenositelnost a také skutečnost, že celý interpret je malý (jádro má pouze 1.4MB).[5]

Stabilní verzi tohoto interpretu lze zdarma stáhnout pod GNU licencí na oficiálních stránkách SWI-Prolog. [5]

## Programování v Prologu

Pro úplnost výkladu je vhodné uvést velmi rychle základy programování v jazyce Prolog. Kompletní návody a výukové materiály lze nalézt například na oficiálních stránkách SWI-Prolog ([5]). Vzhledem k tomu, že se jedná o logický programovací jazyk, lze předpokládat, že programování v tomto jazyce bude vypadat jinak, než u běžných procedurálních jazyků. Program napsaný v Prologu je zapsán pomocí klauzulí. Jak bylo řečeno, čistý Prolog se omezuje pouze na predikátovou logiku prvního řádu (konkrétně Hornovy klauzule). Databázi klauzulí lze potom rozdělit na fakta a pravidla. Faktem rozumíme skutečnost, která je považovaná za pravdivou. V Prologu lze zapsat následující fakt:

`student(anna) .`

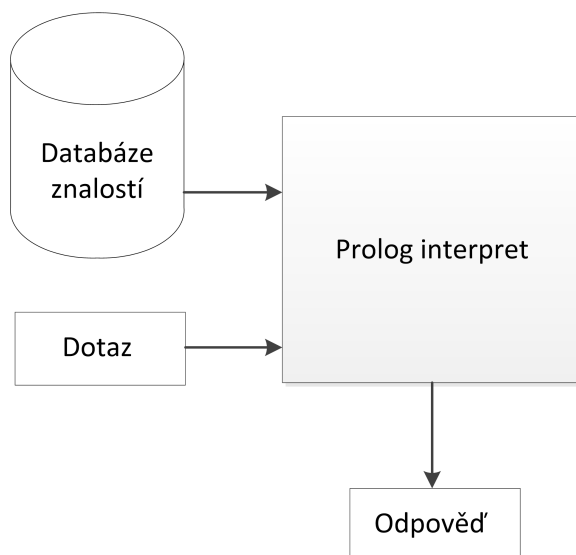
Tímto faktem rozumíme, že Anna je studentem nějaké školy. Množina takových faktů tvoří databázi znalostí. Na tento fakt je možné se dotázat zadáním dotazu `student(anna)`. Interpret Prologu provede vyhodnocení, tedy prohledá databázi znalostí a odpoví jestli je daný dotaz pravdivý anebo ne. Dotaz lze formulovat také následovně: `student(X)`, kde X je proměnná. Interpret se potom pokusí unifikovat fakty z databáze znalostí s proměnnou X. Výsledkem bude množina alternativních unifikací, kdy vlastně interpret nalezne všechna jména osob, které jsou studenty. Stejně tak se lze dotazovat na pravidla. Pravidlo může mít například následující tvar:

`inzenyr(X) :- student(X), absolvent(X) .`

Toto pravidlo říká, že inženýrem je osoba, která je studentem a zároveň je úspěšným absolventem. Pokud bychom dotaz formulovali následovně: `inzenyr(anna)`, interpret Prologu provede SDL rezoluci, tedy prohledávání stavového prostoru do hloubky. Pokud dojde

k úspěšnosti všech podcílů, odpoví **true**. V případě selhání některého z podcílů odpoví **false**. Opět lze dotaz formulovat také jinak: `inzenyr(X) .`, a výsledkem vyhodnocení tohoto dotazu budou všechny osoby, které jsou inženýry. Skládáním takovýchto dotazů lze vytvářet rozsáhlé a velmi efektivní programy. [11].

Zjednodušeně lze dotazování v Prologu chápat tak, jak je znázorněno na obrázku 2.13



Obrázek 2.13: Diagram dotazování se do databáze znalostí v Prologu.

## Kapitola 3

# Analýza požadavků a návrh systému

Důležitou součástí vývoje každého projektu je důkladná analýza požadavků a návrh aplikace. V této kapitole je popsán profil očekávaného uživatele a jsou definovány požadavky na základní funkcionalitu programu. Dále je podrobně popsána struktura vstupních dat, se kterými bude vyvíjený systém pracovat. V závěru kapitoly je věnován prostor popisu pravidel, která jsou platná v genealogických strukturách, a jejichž platnost bude systém schopen ověřovat.

### 3.1 Popis uživatele a přehled očekávaných funkcí

Tato aplikace předpokládá jako svého uživatele odborníka v oblasti genealogie a informatiky. Jako genealog by měl uživatel mít k dispozici genealogickou databázi, se kterou chce dále pracovat. Předpokládá se také jeho znalost genealogických pojmů. Vzhledem ke skutečnosti, že nebude implementováno grafické uživatelské rozhraní, je zapotřebí alespoň základní znalost jazyka Prolog. Veškeré dotazování bude probíhat přímým zadáváním příkazů v prostředí SWI-Prolog. Uživatel by tedy měl být schopen prostudovat manuál a porozumět používání jednotlivých dotazů. Stejně tak je znalost programování v jazyce Prolog potřebná pro vytváření nových dotazů.

Ze zadání lze odvodit řadu očekávaných funkcí. Následuje seznam funkcí, které budou implementovány ve výsledném programu.

- Import vlastních genealogických dat
- Přidávání záznamů za běhu programu
- Export upravené databáze do souboru
- Kontrola konzistence genealogických záznamů
- Vytváření statistik z genealogické databáze
- Vytváření genealogických struktur
- Vyhledání konkrétního vztahu mezi osobami



## 3.2 Prvotní analýza požadavků

Ze zadání lze vyvodit základní požadavky, které by výsledný systém měl splňovat. Systém by měl definovat celou řadu omezení, která jsou platná pro vztahy mezi jednotlivými osobami. Tato pravidla jsou uvedena v kapitole 3.7. S pomocí takovýchto omezení lze validovat vztahy popsané v genealogických datech. Systém by měl být schopen ověřit platnost dodané genealogické struktury. Použitím stejných pravidel by pak mělo být možné z dodaného přepisu nějaké matriční knihy zreprodukovat genealogickou strukturu. Tím se dostáváme k dalšímu požadavku, a tedy, že pro zadanou osobu/rod by systém měl být schopen sám vytvořit genealogickou strukturu. Vzhledem k nejednoznačnosti některých vztahů by pak mohlo dojít k různým možnostem organizace takové struktury. Systém by měl vyhodnotit všechny možné eventuality a poskytnout uživateli na výběr z možností.

Posledním významným požadavkem je možnost uživatelského vytváření dotazů. Toto lze docílit důkladnou analýzou problému a vytvořením jednotného formátu výstupů implementovaných dotazů. Navržením vhodného standardu pro vzhled dat by pak bylo možné kombinovat existující dotazy a vytvářet tak nové. Jazyk Prolog takové chování podporuje a tento požadavek je tedy splnitelný.

## 3.3 Use case diagram a systémový sekvenční diagram

Pro lepší představu funkcionality výsledného programu popsané v kapitole 3.4 byl vytvořen diagram případů užití, neboli use-case diagram (příloha B.1). Tento diagram zobrazuje přehled operací, které může se systémem uživatel provádět. Vzhledem k charakteru systému není potřeba přidělovat uživatelům různé role. V diagramu případů užití nejsou zahrnuty reakce systému, proto byl vytvořen také systémový diagram sekvence (příloha C.1), ve kterém jsou zahrnuty odpovědi systému.

## 3.4 Funkční požadavky

Dle výčtu očekávaných funkcí výsledného systému, uvedeného v kapitole 3.1, lze nyní definovat funkční požadavky. Ty jsou uvedeny v tabulce 3.1. Funkční požadavky popisují chování systému a jeho interakci s okolím. Pro každou očekávanou funkci je v tabulce popsán požadovaný vstup, jeho zpracování a očekávaný výstup. Taková analýza poskytuje detailnější přehled o funkcionality systému.

## 3.5 Další požadavky

Pro zabezpečení co nejvyšší přenositelnosti výsledného programu bude využit volně dostupný interpret jazyka Prolog a to SWI-Prolog (kap. 2.9). Vzhledem k jeho vysoké kompatibilitě s většinou operačních systémů byl vybrán jako nejvhodnější.

S tím je spojena také jednoduchá instalace výsledného programu, kdy uživatel jednoduše zdarma stáhne a nainstaluje interpret SWI-Prolog a potom již stačí pouze otevřít výsledný program v tomto prostředí.

Systém, který bude výsledkem této práce nebude vyžadovat žádné nadstandardní hardwarové vybavení počítače.

	<b>Vstup</b>	<b>Zpracování</b>	<b>Výstup</b>
<b>Import databáze</b>	Uživatel zadá název souboru, ve kterém je databáze uložena.	Soubor je otevřen a databáze je načtena do vnitřní databáze znalostí v prostředí SWI-Prolog.	Databáze byla úspěšně importována do prostředí SWI-prolog. Pokud vstupní soubor neexistuje anebo jej nelze otevřít, je na tuto skutečnost uživatel upozorněn.
<b>Přidání záznamů za běhu programu</b>	Uživatel zadá všechny požadované parametry nového záznamu.	Je ověřeno zachování konzistence databáze.	Je zobrazen dotaz, jestli chce uživatel skutečně vložit nový záznam. Pokud se vyskytlo porušení konzistence, je na tuto skutečnost uživatel upozorněn.
<b>Export upravené databáze do souboru</b>	Uživatel zadá jméno souboru a typ databáze, která má být exportována.	Jsou vyhledány odpovídající záznamy a jsou uloženy do souboru.	Databáze byla exportována do souboru. Pokud operace selže, anebo není možné vytvořit soubor, je na tuto skutečnost uživatel upozorněn.
<b>Kontrola konzistence genealogických záznamů</b>	Uživatel zadá požadavek na kontrolu konzistence databáze.	Jsou provedeny odpovídající kontroly konzistence.	Uživatel je obeznámen s výsledkem kontroly.
<b>Vytváření statistik z genealogické databáze</b>	Uživatel zadá požadavek na výpočet statistik.	V databázi jsou vyhledány odpovídající záznamy a je proveden výpočet.	Statistika je zobrazena na výstupu ve formě textu.
<b>Vytváření genealogických struktur</b>	Uživatel zadá požadavek na sestavení rodové linie určité osoby.	Je provedeno vyhledání odpovídajících předků.	Seznam osob, které patří do takové struktury.
<b>Vyhledání konkrétního vztahu mezi osobami</b>	Uživatel zadá konkrétní osoby, mezi kterými má být vyhledán jejich vzájemný vztah	Je provedeno vyhledání odpovídajících záznamů a je vypočítán vztah mezi osobami.	Výsledek je zobrazen. Pokud mezi osobami vztah neexistuje, je na tuto skutečnost uživatel upozorněn.

Tabulka 3.1: Funkční požadavky

Vzhledem k zadání této práce nebude implementováno grafické uživatelské rozhraní. Uživatelské rozhraní systému bude pouze v podobě příkazového řádku v prostředí SWI-Prolog. Tento systém by měl tvořit logickou vrstvu a vytvoření uživatelského grafického rozhraní je ponecháno pro další vývoj.

Přesnost výsledků výsledného programu je zcela závislá na kvalitě a úplnosti dodané genealogické databáze. Pro správné fungování je nutné, aby uživatel dodal data ve správné struktuře (popsána v následující kapitole). Dále je nutné, aby databáze obsahovala všechny typy genealogických záznamů (tzn. záznamy o narozených, zemřelých a o sňatcích).

### 3.6 Specifikace struktury dat

Systém předpokládá konkrétní strukturu vstupních dat. Vzhledem k charakteru této práce je vhodné reprezentovat data v podobě co nejvíce podobné samotným záznamům v genealogických knihách. Data budou odpovídat záznamům v matričních knihách, tedy na každém řádku budou uvedeny informace o jedné osobě. Ne každý záznam v knize narození obsahuje všechny informace. Především nejstarší záznamy, které pochází z první poloviny osmnáctého století, obsahují pouze základní informace. Záznamy buď nejsou kompletní, případně nejsou čitelné, nebo se nedochovaly. Původní data, která slouží jako podklad k této práci, byla přepisována ručně z matričních knih do tabulek v programu Microsoft Excel. Takovou tabulku lze jednoduše exportovat do souboru `.csv`. Každý řádek tohoto souboru potom představuje jednu relaci, kterou lze v jazyce Prolog reprezentovat jako fakt. V následujících podkapitolách je popsána struktura záznamů z jednotlivých matričních knih.

Zajímavou informací je číslo rodiny. Záznamu o sňatku dvou lidí je přiřazeno číslo rodiny. Potom všichni potomci tohoto páru jsou vždy označeni stejným číslem rodiny, díky čemu lze rozlišit kdo je a kdo není členem rodiny. V případě opětovného sňatku některého z manželů (v případě rozvodu anebo úmrtí partnera) získá nové číslo rodiny. Pro vypracování této práce byla dodána takto strukturovaná data od vedoucího této práce p. doc. Zbořila. Přítomnost čísla rodiny se tedy předpokládá.

#### Data o narozených

Struktura dat o narozených osobách dle dostupných prepisů matričních knih:

1	Číslo zápisu
2	Číslo strany v matriční knize
3	Číslo provázání (zachováno pro další rozšíření)
4	Standardizovaný tvar jména rodiny
5	Číslo rodiny
6	Měsíc narození
7	Den narození
8	Rok narození
9	Křestní jméno narozeného
10	Příjmení narozeného
11	Křestní jméno otce
12	Povolání otce
13	Křestní jméno matky
14	Místo narození
15	Číslo popisné

Informace o narozených osobách budou uloženy v databázi v podobě faktů `mrecord/15`.  
Příklad takového faktu:

```
mrecord(19,2698,1,'KUZMIC',19,11,5,1686,'Martin','Kuzmitch','Paolo','judex',
'Dorothea','Frl',2).
```

## Data o manželstvích

Struktura dat o svatbách dle dostupných prepisů knih manželství:

1	Číslo zápisu
2	Číslo strany v knize manželství
3	Číslo provázání (zachováno pro další rozšíření)
4	Číslo rodiny
5	Měsíc konání svatby
6	Den konání svatby
7	Rok konání svatby
8	Jméno ženicha
9	Příjmení ženicha
10	Jméno otce ženicha
11	Povolání otce ženicha
12	Místo narození
13	Číslo popisné
14	Jméno nevěsty
15	Příjmení nevěsty
16	Jméno otce nevěsty
17	Povolání otce nevěsty
18	Místo narození
19	Svědkové
20	Poznámka

Informace o oddaných osobách budou uloženy v databázi v podobě faktů `marrecord/20`.  
Příklad takového faktu:

```
marrecord(67,3,2,171,21,1,1691,'Joannes','Bela','Nicola','judex','Frl',6,
'Helena',nonym,'Viti','sartor','Bisztrice','Oroszvanic','Mathio Babich,
'Jacoby Vuk',").
```

## Data o zemřelých

Pro zaznamenání rodinného stavu zemřelého se ve starších matrikách používaly latinské názvy takového stavu. Číslo rodiny potom odpovídá tomuto stavu (pokud zemřela osoba jako něčí dcera, je číslo rodiny stejné jako v záznamu o jejím narození. Pokud zemře jako manželka, číslo rodiny odpovídá číslu rodiny které jí bylo přiděleno v záznamu o svatbě.) V databázi se pro vyjádření takové informace používají odpovídající zkratky:

- vd - vidua (vdova)
- vid - vidus (vdovec)
- fi, fl - filius/ia (syn/dcera)
- ux - uxor (manželka)
- cs - consorte (manžel)

Struktura dat o úmrtích dle dostupných přepisů knih zemřelých:

1	Číslo záznamu
2	Číslo strany v knize zemřelých
3	Standardizované příjmení
4	Číslo rodiny
5	Měsíc úmrtí
6	Den úmrtí
7	Rok úmrtí
8	Rodinný stav zemřelého při úmrtí (dítě, manžel, vdova, ...)
9	Příjmení zemřelého
10	Jméno zemřelého
11	Jméno otce zemřelého
12	Povolání
13	Místo narození
14	Věk v letech
15	Věk v měsících (v případě úmrtí dítěte do jednoho roku)
16	Věk ve dnech (v případě úmrtí dítěte do jednoho roku)
17	Příčina

Informace o zemřelých budou uloženy v databázi v podobě faktů `deathrecord/17`. Příklad takového faktu:

```
deathrecord(6,12,'CERNY',22,6,6,2001,'cs','Cerny','Igor','Martin','judex',
'frl',56,0,0,'nemoc').
```

## Řešení různých tvarů stejného příjmení

Častým jevem objevujícím se v genealogických záznamech je, že jedno příjmení je v různých záznamech zapsáno různě. Příjmení Černý se může vyskytovat jako Czerny, Schwartz, Cherni, Schvartz apod. Tyto odlišnosti mají mnoho zdrojů. Ke změnám příjmení často vedla společenská či politická situace (poněmčování jmen za druhé světové války), anebo prostá chyba matrikáře. Aby bylo možno takové záznamy strojově zpracovávat, je potřeba data vhodně předzpracovat. První myšlenkou bylo zavedení standardizovaného jména, které by bylo uvedeno u každého záznamu a označovalo by tak osoby stejného příjmení. Přidávat tuto informaci ke každému záznamu by však bylo zbytečně pracné a zvyšovalo by to redundanci. Proto byla zavedena struktura `namelist/2`, která slouží jako překladová tabulka. Prvním parametrem je zmíněné standardizované příjmení, druhým parametrem je potom seznam všech variant příjmení, která se v záznamech objevují. Pro uvedený příklad příjmení Černý by taková struktura vypadala následovně:

```
namelist('CERNY', ['Czerny','Schwartz','Cherni','Schvartz']).
```

### 3.7 Pravidla platná ve vztazích

Jak bylo již řečeno v kapitole 2.5, každá genealogická struktura popisuje vzájemné vztahy mezi osobami. Tyto vztahy se od sebe navzájem liší množinou pravidel, která vztah definují. Takových pravidel existuje celá řada a jejich nalezení a definice je nezbytná pro tuto práci. Následuje seznam takových pravidel, která budou ověřována ve výsledném programu:

- Manželé nesmí být sourozenci
- Mezi manželi nesmí být vztah rodič-potomek
- Svatba nesmí nastat před narozením kteréhokoliv z manželů.
- Svatba nesmí nastat po smrti kteréhokoliv z manželů.
- Sourozenci od stejných rodičů se mohou narodit s odstupem minimálně 9 měsíců, anebo se jedná o dvojčata se stejným datem narození
- Sourozenci by neměli být narození s odstupem více než 30 let.
- Sourozenci nesmí mít stejné křestní jméno
- Dítě se může narodit pouze živé matce
- Muž se může stát otcem dítěte nejpozději 9 měsíců po smrti
- Nevěsta/ženich by měli v den svatby mít alespoň 15 let
- Otec/matka by měli být alespoň o 15 let starší než jejich potomek
- Matka by neměla být starší než její potomek o více jak 50 let
- Prarodič by měl být alespoň o 30 let starší než potomek (vychází z tranzitivity vztahů)

Podobně by bylo možné definovat další pravidla, například, že rodič a prarodič nesmí být sourozenci, atd. Taková pravidla však není třeba uvádět, protože plynou z tranzitivních vlastností genealogických struktur (kap. 2.6).

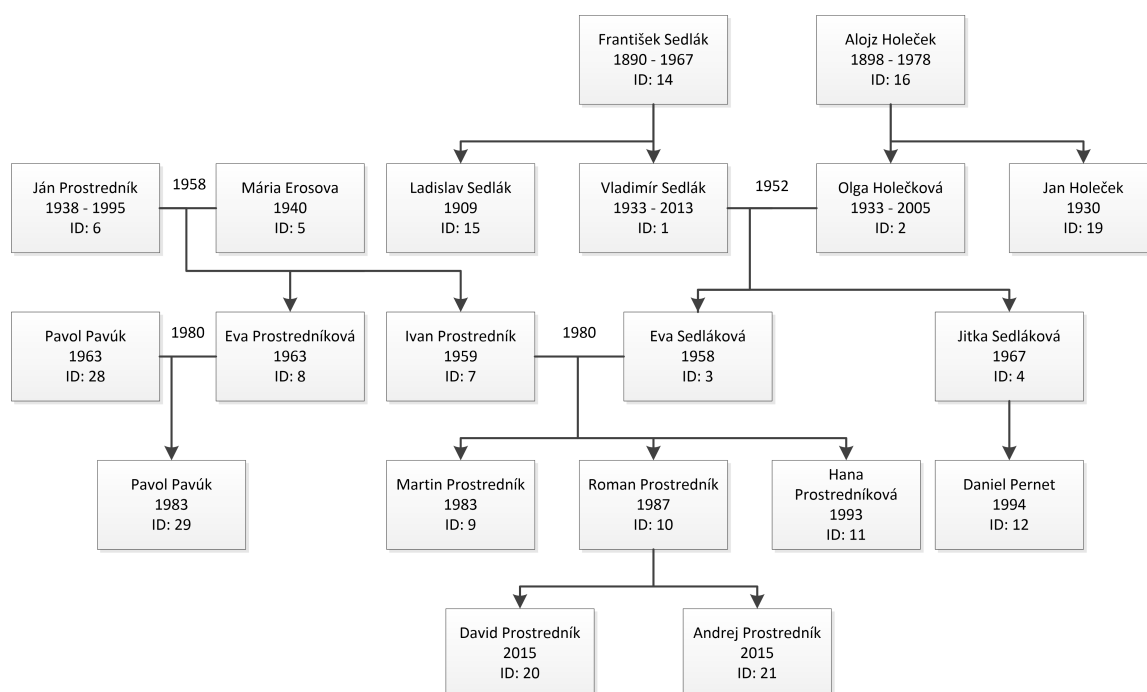
Je patrné, že některá pravidla lze definovat zcela přesně. S absolutní jistotou můžeme prohlásit, že potomek musí být mladší než jeho rodič. Není v žádném případě možné, aby toto pravidlo nebylo platné. Jiná pravidla však jsou definována pouze odhadem. Například, že věkový rozestup mezi matkou a potomkem by neměl být větší než 50 let. Jen velmi výjimečně se stane více než padesátiletá žena matkou (dle [19] bylo v r. 2015 v České republice narozeno 13 živých dětí ženám, které byli starší než padesát let).

Tato pravidla budou následně využita při kontrole konzistence databáze. Stejně tak hrají tato pravidla důležitou roli při rekonstrukci genealogických struktur.

## Kapitola 4

# Implementace výsledného systému

Dle specifikací a návrhu byl implementován výsledný systém. Program byl implementován v prostředí SWI-Prolog, které je pro uživatele dostupné zdarma. V následujících podkapitolách jsou podrobně popsány jednotlivé části výsledného programu. U každé části je vysvětlen způsob implementace, doplněn o ukázkou použití konkrétního dotazu. Při vývoji se pracovalo s testovacími daty, jejichž struktura je zobrazena na obrázku 4.1.



Obrázek 4.1: Struktura ukázkových dat použitých při vývoji systému.

### 4.1 Struktura programu

Vzhledem k rozsahu zadání bylo nezbytné při implementaci dodržet logické dělení kódu. Logicky související části kódu jsou implementovány v rámci jednoho souboru. Význam jednotlivých souborů je popsán v tabulce 4.1.

<code>main.pl</code>	Jedná se o základ celého programu. Je zde proveden import všech ostatních zdrojových kódů a genealogické databáze. Dále soubor obsahuje dotazy nezbytné pro přidání dalších genealogických záznamů.
<code>import.pl</code>	Tento malý soubor slouží pro import genealogické databáze. Uživatel v tomto souboru změní názvy souborů dle svých vlastních, případně zakomentuje nežádoucí řádky (např. v případě, že veškeré záznamy má uloženy v jednom jediném souboru). Další způsoby vkládání genealogických záznamů jsou uvedeny v kapitole 4.2.
<code>export.pl</code>	Zde jsou implementovány veškeré dotazy, které jsou potřebné pro export upravené genealogické databáze do souboru ve formátu <code>.pl</code> anebo <code>.csv</code> . Detailnější popis je uveden v kapitole 4.7.
<code>usage.pl</code>	Soubor obsahující manuál k používání jednotlivých dotazů.
<code>borns.pl</code> <code>marriages.pl</code> <code>deaths.pl</code>	V těchto souborech se nacházejí veškeré dotazy, které jsou potřebné k vytvoření statistik o narozených, zemřelých a o sňatcích. Výpočet statistik je podrobně popsán v kapitole 4.4.
<code>messages.pl</code>	Soubor obsahující všechna informační či chybová hlášení, která jsou výsledkem implementovaných dotazů.
<code>facts.pl</code>	V tomto souboru se nacházejí definice pravidel, která musí být platná v genealogických strukturách. Pravidla odpovídají pravidlům uvedeným v kapitole 3.7.
<code>check.pl</code>	Soubor, ve kterém jsou obsaženy dotazy týkající se kontroly konzistence genealogické databáze. Význam a implementace jednotlivých dotazů je popsán v kapitole 4.3.
<code>lines.pl</code>	Jedná se o klíčový soubor obsahující implementaci dotazů, které vyhledávají mužskou či ženskou rodovou linii. Více o konkrétní implementaci je napsáno v kapitole 4.5.
<code>relations.pl</code>	Soubor obsahující dotazy pro vyhledání vztahu mezi dvěma osobami. Implementace popsána v kapitole 4.6.
<code>manipulate.pl</code>	Tento soubor obsahuje pomocné dotazy, které zabezpečují úpravu dat a ostatní manipulaci s daty.
<code>name.pl</code>	Soubor sloužící jako překladová tabulka pro převod příjmení na jeho standardizovanou podobu.

Tabulka 4.1: Struktura výsledného programu.

## 4.2 Vložení vstupních dat

Program předpokládá genealogická data, která přísně splňují formát popsáný v kapitole 3.6. Jak bylo řečeno, jeden záznam odpovídá jednomu predikátu jazyka Prolog. Čím kvalitnější data budou, tím spolehlivěji bude program pracovat. Správný import genealogické databáze je tedy velmi důležitý. Záznamy lze vkládat několika způsoby. První možností je ruční úprava souboru `import.pl`, který má následující podobu:



```

%database narozenych
:-include('test.pl').
%database manzelstvi
:-include('test_marr.pl').
%database zemrelych
:-include('test_dead.pl').

```

Uživatel jednoduše změní názvy souborů na své vlastní. Spuštěním programu je potom automaticky provedeno načtení záznamů z odpovídajících souborů. Tím je v prostředí SWI-Prolog vytvořena databáze, nad kterou lze provádět dotazování. Není nutné aby byly záznamy rozděleny do tří souborů. Všechny typy záznamů mohou být uloženy v jednom jediném souboru, důležité však je, aby soubor obsahoval všechny typy záznamů (tedy záznamy o narozeních, manželstvích, a úmrtích) pro správnou funkcionální systém.

Dalším způsobem vložení dat je implementovaný dotaz `add_database/1`, kterým lze ručně importovat data ze souboru počas běhu prostředí SWI-Prolog. Kdykoliv během používání programu tedy lze přidat další genealogické záznamy.

Poslední možností je ruční vkládání jednotlivých záznamů. Tento způsob je vhodný, pokud chce uživatel doplnit jen malý počet záznamů. Proto byly implementovány dotazy `add_born_record/12`, `add_marriage_record/19` a `add_death_record/15`. Při vkládání jednoho nového záznamu je provedena kontrola konzistence dat (podrobněji popsáno v dalších kapitolách). Pokud nově vložený dotaz porušuje pravidla konzistence dat, je na tuto skutečnost uživatel upozorněn a program se zeptá, jestli má být záznam skutečně vložen. Pokud se uživatel rozhodne nerespektovat upozornění a přesto povolí záznam, je tento záznam vložen. Později je při každé uživatelem vynucené kontrole konzistence uvedeno varování na špatně vložená data.

### 4.3 Ověření konzistence dat

Jádrem výsledného programu je ověření konzistence uživatelem vložené genealogické databáze. Toho je docíleno kontrolou platnosti pravidel, uvedených v kapitole 3.7. Tato pravidla jsou všeobecně platná a tedy musí být splněna v jakékoliv genealogické struktuře. Veškeré dotazy pro kontrolu konzistence jsou implementovány v souboru `check.pl`.

Před spuštěním kterékoliv z implementovaných kontrol je nejdříve zkontrolována dostupnost všech typů genealogických záznamů. Dotaz `check_databases` provede kontrolu, jestli byly importovány záznamy o narozených, oddaných i zemřelých. Pokud některý typ záznamu zcela chybí, je zobrazeno chybové hlášení a kontrola dále nepokračuje, protože pro další kontroly je nezbytná přítomnost všech typů záznamů.

Následuje kontrola jedinečného identifikátoru. Dotaz `check_id/0` provede kontrolu, zda má každý záznam jedinečné identifikační číslo. Tato identifikace musí být jedinečná pouze v rámci konkrétního typu záznamu (je tedy přípustné, aby v databázi existoval například záznam o narození se stejným identifikátorem jako by měl záznam o úmrtí). Tato kontrola může být provedena pro každý typ databáze samostatně. K tomu slouží příkazy `check_id_born/0` pro záznamy o narozených, `check_id_marriage/0` pro záznamy o manželstvích a pro záznamy o zemřelých `check_id_death/0`. Ukázka výsledku takové kontroly je vyobrazena na obrázku 4.2.

```
?- check_id.  
Error: Identification numbers are not unique in born records.  
OK: Check of IDs in marriage records  
OK: Check of IDs in deaths records  
true.
```

Obrázek 4.2: Ukázka výsledku kontroly jedinečnosti identifikátorů záznamů.

Jako další je provedena kontrola rodičovských párů. Jsou vyhledáni všichni potomci konkrétní rodiny (rodina definována číslem rodiny) a je prohledána databáze zemřelých, ve které jsou vyhledáváni jejich rodiče. Pokud databáze obsahuje záznam o smrti matky, je ověřeno, jestli je datum úmrtí nejdříve v den narození potomka (v případě úmrtí při porodu). V případě otce může být potomek narozen až devět měsíců po jeho smrti. K této kontrole slouží dotaz `check_parents_alive/0`. Pro kontrolu pouze otců a pouze matek slouží dotazy `check_father_alive/0` a `check_mother_alive/0`. Pokud jsou nalezeny záznamy, které nesplňují zadaná kritéria, jsou informace o nich vypsány v podobě chybového hlášení. Ukázka takové kontroly je zobrazena na obrázku 4.3.

```
?- check_parents_alive.  
OK: All fathers alive when descendants were born.  
  
Error: Descendant born after mothers death:  
    Mother death: Olga Sedlakova - 26.8.2005 (death record 2)  
    Descendant born: Eva Sedlakova - 5.9.2006 (born record 3)
```

Obrázek 4.3: Ukázka výsledku kontroly úmrtí rodičů.

Následuje kontrola, jestli osoba může být podle svého věku rodičem svých dětí. Postupně jsou kontrolovány všechny osoby. Nejdříve je kontrolována matka, u které se předpokládá, že bude minimálně o 15 let a maximálně o 50 let starší. Horní hranice je mírně nadsazena pro extrémní případy. Následně jsou kontrolováni otcové, kde se předpokládá pouze to, že musí být starší alespoň o 15 let než jeho potomek (lze předpokládat, že muž se může stát otcem v jakémkoliv věku). Pro takovou kontrolu byl implementován dotaz `check_parents/0`. V případě nalezení záznamů, které nesplňují daná pravidla je na tuto skutečnost uživatel upozorněn chybovým hlášením. Na obrázku 4.3 je zobrazena ukázka takovéto kontroly.

```
?- check_parents.  
Error: Mother isn't the right age.  
    Mother: Eva Sedlakova born in 1970. (record 3)  
    Descendant: Martin Prostrednik born in 1983. (record 9)  
  
OK: All fathers are of the correct age.
```

Obrázek 4.4: Ukázka výsledků kontroly správnosti věku rodičů.

Dalším typem kontroly konzistence databáze je kontrola manželů. Dle pravidel manželé nesmí být sourozenci. Při takto navržené struktuře dat to znamená, že manželé nesmějí mít stejné číslo rodiny. Dále platí, že manželé musí být v den svatby starší patnácti let. Rovněž je kontrolováno datum svatby a porovnává se s daty narození obou novomanželů. Svatba se nemohla stát před jejich narozením a stejně tak nemohla nastat ani po jejich smrti. Kontrola

manželských párů se provádí dotazem `check_spouse/0`. Ukázka výsledku takové kontroly je uvedena na obrázku 4.5.

```
?- check_spouses.  
Error: Marriage in 1952 happened after spouses death.  
      Olga Holeckova died in 1905 (death record 2)  
  
Error: Spouses are too young for marriage in 1954:  
      Jan Prostrednik born in 1938 (record 6)  
      Maria Erosova born in 1940 (record 5)  
  
Error: Marriage in 1985 happened before spouse was born.  
      Roman Prostrednik born in 1987 (record 10)  
  
true.
```

Obrázek 4.5: Ukázka výsledků kontroly manželských párů.

Následně jsou kontrolována pravidla platící pro sourozence. Platí, že dva sourozenci musí být narozeni s odstupem alespoň devíti měsíců. Stejně tak se nepředpokládá rozdíl mezi pokrevními sourozenci větší než třicet let. Jedinou výjimkou je narození dvojčat, kdy mají oba sourozenci stejné datum narození. Kontrolu těchto pravidel zabezpečuje dotaz `check_siblings_age/0`. Dále jsou kontrolována křestní jména sourozenců, která nesmějí být stejná. O kontrolu tohoto pravidla se stará dotaz `check_siblings_name/0`. Souhrnně lze pravidla platná pro sourozence kontrolovat dotazem `check_siblings/0`. Pokud je porušena platnost kteréhokoliv pravidla, je zobrazeno chybové hlášení. Ukázka takové kontroly je na obrázku 4.6.

```
?- check_siblings.  
OK: All siblings birth dates are valid.  
  
Error: Siblings have same first name:  
      28.5.2015 David Prostrednik (record 111)  
      30.5.2016 David Prostrednik (record 21)  
  
true.
```

Obrázek 4.6: Ukázka výsledků kontroly sourozenců.

Kompletní kontrolu lze provést dotazem `complete_check/0`, který postupně provede všechny uvedené kontroly. Za zmínku stojí také skutečnost, tento dotaz byl upraven tak, aby bylo možno kontrolovat pouze osoby konkrétního příjmení.

## 4.4 Statistiky

Po tom co uživatel nechá provést ověření, jestli jsou všechny vztahy v genealogické databázi korektní, může začít s daty pracovat. První funkcionalitou, která byla implementována, je možnost zobrazení statistik o jednotlivých typech genealogických databází. Takové statistiky mohou mít vysokou výpovědní hodnotu o prostředí a době, ze které databáze pochází. Lze pozorovat rozdíly v průměrné délce života, či v příčinách smrti. Dále je zajímavý prů-

měrný počet dětí v rodině a kolik z nich se dožilo dospělosti. Význam takových statistik je z historického hlediska velmi velký a vypovídá o životních podmínkách v konkrétní době.

Dotazy potřebné pro výpočet statistik o narozených jsou implementovány v souboru `borns.pl`. V souboru `marriages.pl` jsou implementovány dotazy pro statistiky o manželstvích a nakonec dotazy pro statistiky o zemřelých nalezneme v souboru `deaths.pl`.

## Statistiky o narozených

Statistiky o narozených lze zobrazit dvěma způsoby. První možností je zobrazit dotazem `born_statistics` souhrnnou statistiku ze všech záznamů o narození. Druhou možností je dotaz dotazem `born_statistics('Prijmeni')`, který zobrazí statistiky osob konkrétního příjmení. V obou případech je sledováno několik údajů. Prvním je informace, kolik záznamů o narození bylo v databázi nalezeno. Dalším je rozsah let, ve kterých byly záznamy zapsány. Následuje informace o roce, ve kterém bylo narozeno nejvíce dětí. Obdobně je uveden také rok, ve kterém se narodilo nejméně dětí. Zajímavým údajem je průměrný počet narozených dětí za sledované časové období. Poslední informací je údaj o nejčastěji dávaném příjmení a křesním jménu. Na obrázku 4.7 je vyobrazen výsledek souhrnné statistiky.

```
?- born_statistics.  
  
Statistics of births  
  
Total born records count: 25  
Recorded years: 1860 - 2015  
The most borns in 1860. Count: 2  
The least borns in 1898. Count: 1  
Average borns count per year: 0.16129032258064516  
The most frequent name is PROSTREDNIK. Count: 8  
The most frequent name is Alojz. Count: 2  
true.
```

Obrázek 4.7: Ukázka souhrnné statistiky o narozených.

## Statistiky o manželstvích

Dotazem `marriage_statistics` lze zobrazit statistiky uzavřených sňatků, které rovněž obsahují množství zajímavých informací. Lze také zobrazit statistiky dle konkrétního příjmení dotazem `marriage_statistics('Prijmeni')`. Tímto příjmením je myšleno příjmení ženicha a tedy i budoucí příjmení nevěsty a jejich potomků. V úvodu jsou opět uvedeny údaje o počtu všech záznamů z databáze manželství a rozsah let, ve kterých byly tyto sňatky uzavřeny. Dále následuje rok ve kterém bylo uzavřeno nejvíce manželství a také rok, ve kterém bylo uzavřeno nejméně sňatků. Uveden je také pár s největším počtem potomků a průměrný počet svateb na jeden rok. Zajímavou informací je průměrný počet dětí připadající na jeden manželský pár. Na obrázku 4.8 je uveden příklad takové statistiky.

```
?- marriage_statistics.

Statistics of marriages

Total marriage records count: 8
Recorded years: 1912 - 2011
The most marriages in 1980. Count: 2
The least marriages in 1912. Count: 1
Average marriages count per year: 0.08080808080808081
The most descendants has couple: Ivan Prostrednik and Eva Sedlakova. Count: 3
Average descendant count per pair: 1.875
true.
```

Obrázek 4.8: Ukázka statistiky o manželstvích.

## Statistiky o zemřelých

Poslední statistikou, kterou lze zobrazit je statistika o úmrtích. Opět lze statistiku zobrazit dotazem `death_statistics` pro všechny záznamy obsažené v databázi, anebo ji dotazem `marriage_statistics('Prijmeni')` vztáhnout pouze k osobám určitého příjmení. Nejprve jsou stejně jako u předchozích statistik uvedeny informace týkající se počtu záznamů a rozsahu let, ve kterých byly záznamy zaznamenány. Dále rok s nejvyšším a nejnižším počtem zapsaných záznamů o úmrtí. Mezi další informace patří průměrný počet úmrtí za rok a také průměrný věk zemřelého. Poslední údaj popisuje kolik procent ze sledovaných záznamů zemřelých tvoří děti do jednoho roku. Na obrázku 4.9 je uvedena ukázka statistiky všech zemřelých.

```
?- death_statistics.

Statistics of deaths

Total death records count: 5
Recorded years: 1965 - 2013
The most deaths in 1965. Count: 1
The least deaths in 1965. Count: 1
Average deaths count per year: 0.10416666666666667
Average death age is: 76.4
Newborns (less than one year) deaths: 0%
true.
```

Obrázek 4.9: Ukázka souhrnné statistiky o zemřelých.

## Souhrnné statistiky

Všechny uvedené statistiky je možné zobrazit najednou. K tomuto slouží dotaz `statistics`, kterým je proveden výpočet a výpis statistik o narozených, manželstvích a úmrtích. Stejně jako u dílčích statistik, i v tomto případě lze dotazem `statistics('Prijmeni')` zobrazit statistiky dle konkrétního příjmení.

## 4.5 Rekonstrukce genealogických struktur

Automatická rekonstrukce genealogických struktur je zajímavý implementační problém. Vzhledem k nejednoznačnosti genealogických dat získaných z matričních knih se nejedná

o banální záležitost. Existuje celá řada genealogických struktur, které jsou popsány v kapitole 2.3. V rámci této práce byla implementována rekonstrukce agnantního a kognantního vývodu, tedy rekonstrukce otcovské a mateřské linie od probanta. Jedná se o základní genealogické struktury, ze kterých lze skládat složitější. V této práci jsou implementovány pro následné vyhledání příbuzenského vztahu mezi dvěma osobami (popsáno v následující kapitole). Všechny dotazy potřebné k rekonstrukci těchto struktur jsou implementovány v souboru `lines.pl`.

## Rekonstrukce otcovské linie

Pro vyhledání otcovské linie byl implementován dotaz `get_masculin_line/4`, který předpokládá jako své parametry křestní jméno, příjmení a rok narození osoby, od které chceme vyhledat otcovskou linii. Implementován byl také dotaz `get_masculin_line/2`, kde je parametrem identifikační číslo osoby. Tento jedinečný identifikátor lze kdykoliv vyhledat dotazem `born_list`, který zobrazí seznam osob s jejich identifikačními čísly. Předpokládá se, že otec má stejné příjmení jako jeho potomek. Algoritmus pracuje následovně:

**Vstup:** Jméno, příjmení a rok narození / identifikační číslo výchozí osoby

1. Podle zadaných informací je nalezen záznam o narození osoby
2. Extrakce čísla rodiny a jména otce z nalezeného záznamu
3. Nalezení záznamu o sňatku otce
4. Extrakce data svatby a jména otce ženicha (tedy děda výchozí osoby)
5. Aplikací pravidel z kap. 3.7 je pomocí těchto informací nalezen záznam o narození otce výchozí osoby
  - (a) Pokud bylo nalezeno více záznamů, je vybrán nejvhodnější otec
6. Nastavení otce jako nové výchozí osoby, pokračuje se bodem 1.
7. Pokud nebyl nalezen záznam o narození otce:
  - (a) Nalezení záznamu o narození dědy výchozí osoby
  - (b) Pokud bylo nalezeno více záznamů je vybrán nejvhodnější děda
  - (c) Děda je nastaven jako nová výchozí osoba a pokračuje se bodem 1.
8. Jinak je algoritmus ukončen.

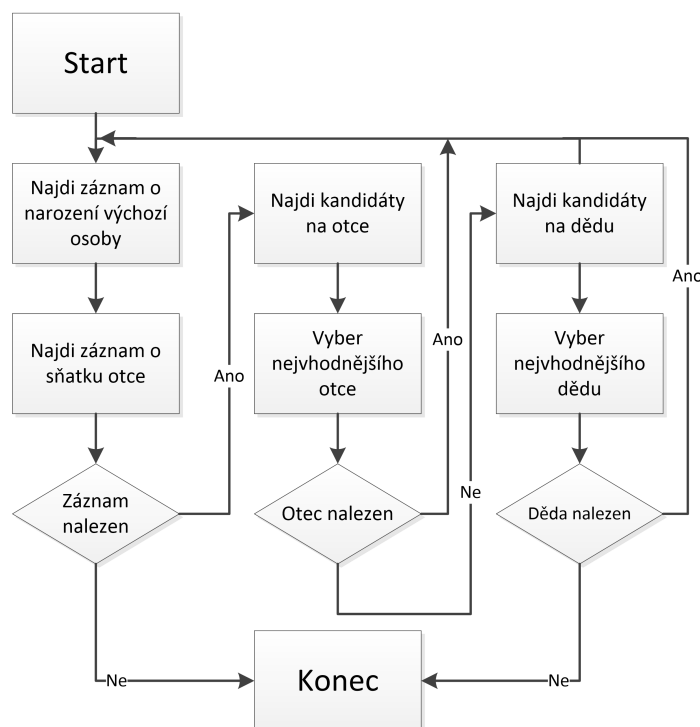
**Výstup:** Zanořený seznam reprezentující otcovskou linii

Na obrázku 4.10 je zobrazen výsledek rekonstrukce mužské linie. Na obrázku 4.11 je zobrazen vývojový diagram znázorňující uvedený algoritmus.

```
?- get_masculin_line('David', 'Prostrednik', 2015,L).  
[[['David','Prostrednik',2015,0,111],[['Roman','Prostrednik',1987,7,10],  
[['Ivan','Prostrednik',1957,6,7],[['Jan','Prostrednik',1938,5,6]]]]]
```

Obrázek 4.10: Ukázka rekonstrukce otcovské linie.

Při vyhledávání může být nalezeno více vhodných kandidátů na otce (např. několik osob stejného jména, jejichž otcové jsou také stejného jména). V takovém případě je z těchto kandidátů vybrán ten nejvhodnější. Výběh nejvhodnějšího otce je řízen pravidly, která musí být platná v genealogických strukturách (kap. 3.7). To znamená, že záznam o narození otce je vyhledán tak, aby byl otec alespoň o patnáct let starší než jeho potomek a aby i v den svatby byl starší patnácti let. Rovněž je kontrolováno číslo rodiny, tak aby potomek a rodič nemohli být manželé. Pokud pravidlům vyhovuje více osob a nelze jednoznačně rozhodnout,



Obrázek 4.11: Vývojový diagram algoritmu pro vyhledání otcovské linie.

kteřá z osob je skutečně otcem, pokračuje vyhledávání pro všechny tyto osoby. Vznikne tak několik možných otcovských struktur, které lze rekonstruovat.

Tento algoritmus se dokáže vyrovnat s absencí záznamu o narození otce, pokud existuje záznam o jeho sňatku. To znamená, že je nalezen záznam o sňatku rodičů výchozí osoby. V případě, že nelze najít záznam o narození otce, případně žádná nalezená osoba nevyhovuje pravidlům platným pro tento vztah, je vyhledán děda výchozí osoby. To je možné díky informaci o jménu otce ženicha v záznamu o uzavření manželství. V případě absence záznamu o sňatku otce výchozí osoby, anebo v případě chybějícího záznamu dvou osob za sebou v otcovské linii nelze tuto linii rekonstruovat. Na obrázku 4.12 je zobrazena rekonstrukce otcovské linie. Je patrná absence informací o otci výchozí osoby.

```

?- get_masculin_line('David', 'Prostrednik', 2015,L).

[[['David', 'Prostrednik', 2015, 0, 111], [], [['Ivan', 'Prostrednik', 1957, 6, 7],
['Jan', 'Prostrednik', 1938, 5, 6]]]]

```

Obrázek 4.12: Ukázka rekonstrukce otcovské linie.

Takto nastavený algoritmus vyžaduje ke své správné funkci jak záznamy o narození tak záznamy o sňatcích. Záznam o uzavření manželství však nemusí být dostupný, ale i přesto databáze obsahuje potřebný záznam o narození otce. Proto byl v rámci experimentu implementován jiný algoritmus, který k rekonstrukci otcovské linie používá pouze záznamy o narození. Spustit jej lze dotazem `get_masculin_line2/4`. Parametry jsou opět křestní jméno, příjmení a rok narození.

Algoritmus nejdříve pro každou osobu uloží do databáze fakt `father/5`, který obsahuje informaci o každé osobě a jejím otci. Následně jsou tyto fakty rekurzivně prohledávány. Pro výchozí osobu je nalezen nejvhodnější otec, kterého databáze obsahuje. Jakmile je nalezen, stává se otec výchozí osobou a vyhledávání se opakuje. Výsledek takového dotazu je na obrázku 4.13.

```
?- get_masculin_line2('David','Prostrednik',2015,L).  
L = ['Roman', 1987, ['Ivan', 1957, ['Jan', 1938, []]]].
```

Obrázek 4.13: Ukázka rekonstrukce otcovské linie pomocí druhého algoritmu.

Jak bylo řečeno, tento algoritmus byl implementován spíše jako experiment a nedosahuje takových výsledků jako předchozí algoritmus. Hlavní nevýhodou je, že jej nelze jednoduše modifikovat k rekonstrukci mateřské linie. Vzhledem k tomu, že záznamy o narození neobsahují rodné příjmení matky a vyhledání matky výchozí osoby je velmi nepřesné (probíhalo by jen podle křestního jména matky a roku narození dítěte). Proto bylo od jeho dalšího vývoje odstoupeno.

## Rekonstrukce mateřské linie

Pro rekonstrukci mateřské linie byli implementovány dotazy `get_femine_line/4`, který předpokládá jako své parametry celé jméno a rok narození a `get_femine_line/2`, který vyhledává dle jednoznačného identifikátoru osoby.

Rekonstrukce mateřské linie probíhá podobným způsobem jako rekonstrukce otcovské linie. Jsou zde však některé podstatné rozdíly. Na obrázku 4.14 je zobrazen výsledek rekonstrukce mateřské linie. Algoritmus pracuje takto:

**Vstup:** Jméno, příjmení a rok narození / identifikační číslo výchozí osoby

1. Podle zadaných informací je nalezen záznam o narození osoby
2. Extrakce čísla rodiny a jména matky z nalezeného záznamu
3. Nalezení záznamu o sňatku matky
4. Extrakce data svatby, rodného příjmení nevěsty a jména otce nevěsty (tedy děda výchozí osoby)
5. Aplikací pravidel z kap. 3.7 je pomocí těchto informací nalezen záznam o narození matky výchozí osoby
  - (a) Pokud bylo nalezeno více záznamů, je vybrána nejvhodnější matka
6. Nastavení matky jako nové výchozí osoby, pokračuje se bodem 1.
7. Pokud nebyl nalezen záznam o narození matky:
  - (a) Nalezení záznamu o narození dědy výchozí osoby
  - (b) Nalezení záznamu o sňatku dědy výchozí osoby
  - (c) Extrakce informací o jeho manželce (tedy o babičce výchozí osoby)
  - (d) Nalezení záznamu o narození babičky + výběr nejvhodnější babičky
  - (e) Babička je nastavena jako nová výchozí osoba a pokračuje se bodem 1.
8. Jinak je algoritmus ukončen.

**Výstup:** Zanořený seznam reprezentující mateřskou linii



```
?- get_femine_line('Hana', 'Prostrednikova', 1993,L).
[[['Hana', 'Prostrednikova', 1993, 7, 11], [['Eva', 'Sedlakova', 1958, 3, 3],
[['Olga', 'Holeckova', 1933, 2, 2]]]]]
```

Obrázek 4.14: Ukázka rekonstrukce mateřské linie.

Obdobně jako při rekonstrukci otcovské linie, i zde je vyhledávání řízeno pravidly platnými pro dané vztahy. Matka tedy musí být alespoň o patnáct let starší než její potomek a neměla by být starší o více než padesát let. Stejně tak nesmí být matka a její potomek manželé. Pokud je nalezeno více osob, které vyhovují daným kritériím a nelze rozhodnout, která z nich je matkou výchozí osoby, pokračuje vyhledávání pro každou osobu. Opět tedy získáváme několik ženských linií, které lze rekonstruovat z dostupných dat.

I v případě vyhledávání mateřské linie je algoritmus schopen vyrovnat se s chybějícím záznamem o narození matky. Je vyhledán záznam o uzavření manželství rodičů výchozí osoby. Pokud na základě takto získaných informací není nalezen záznam o narození matky, případně žádný záznam nevyhovuje pravidlům, je vyhledána babička výchozí osoby. V tomto bodě se algoritmus liší od algoritmu pro vyhledávání mužské linie. Není nám totiž známo rodné příjmení babičky a tedy není možné přesně vyhledat osobu, která by ní mohla být. Proto je vyhledán záznam o narození matčina otce (děda výchozí osoby) a následně záznam o jeho sňatku. Zde je již uvedeno jak křestní jméno, tak rodné příjmení babičky a také jméno jejího otce. Pomocí toho lze následně vyhledat záznam o jejím narození. Tímto způsobem může být rekonstruována mateřská linie v případě jednoho chybějícího záznamu o narození.

## Problémy rekonstrukce genealogických struktur

Největším problémem, který se týká všech operací s genealogickými daty, je nízká kvalita těchto dat. Uvedený algoritmus předpokládá existenci jak záznamů o narození, tak záznamů o uzavření manželství. Velmi lehce se může stát, že při genealogickém výzkumu nejsou nalezeny všechny potřebné informace, případně tyto informace nejsou dostupné vůbec. Algoritmus tedy pracuje natolik spolehlivě, nakolik jsou kvalitní vstupní data. Velkým problémem jsou také obměny křestních jmen v jednotlivých záznamech. Křestní jméno se totiž může lišit v záznamu o narození a v záznamu o manželství, a to například chybou matrikáře, případně je jméno z důvodů současné politické situace pozměněno (např. je zapsána jeho poněmčená verze). Vyjímkou nejsou ani překlapy. Takové chyby lze korigovat jen těžko, a správná funkce algoritmu je podmíněna řádným předzpracováním vstupních dat.

## 4.6 Vyhledání vztahu mezi dvěma osobami

Možnost ověřit správnost genealogických dat a rekonstruovat z nich genealogické struktury nám přináší další možnosti jejich zpracování. Jako první se nabízí možnost vyhledání vztahu mezi dvěma osobami. V rámci vypracování této práce byli implementovány dva různé přístupy k této problematice. Všechny dotazy potřebné k vyhledání vztahu mezi dvěma osobami jsou implementovány v souboru `relations.pl`.

### Algoritmus č.1

První přístup je založen na postupném porovnávání otcovských a mateřských linií dvou osob. Implementuje jej dotaz `relation/7`, jehož parametry je jméno, příjmení a rok naro-

zení obou osob, mezi kterými chceme vyhledat vztah. Implementována byla také varianta `relation/3`, kde jsou osoby zadány jejich identifikačním číslem.

Algoritmus pracuje následovně. Od každé osoby je vyhledána otcovská linie. Tyto linie jsou potom navzájem porovnávány. V případě pokrevního příbuzenstva se totiž vždy dostaneme do bodu, kdy dvě osoby mají společného jednoho či více předků. Proto pokud jsou otcovské linie identické (až na první záznam, kterým je výchozí osoba), jedná se o sourozensce. Pokud se otcovská linie druhé osoby rovná otcovské linii první osoby bez prvního záznamu, jedná se o vztah otec-potomek, a pod. Porovnávání probíhá do třetí generace od výchozí osoby (tedy po pra-pra rodiče). V případě, že byl nalezen odpovídající vztah, je vyhledávání ukončeno. Pokud nebyl vztah nalezen v otcovské linii, je vyhledána mateřská linie každé zadané osoby a tyto linie jsou porovnány stejným způsobem.

Jestliže nebyl nalezen vztah mezi zadanými osobami, algoritmus vezme jako novou výchozí osobu otce první osoby a rekurzivně vyhledává vztah mezi otcem první zadané osoby a druhé osoby. Jestliže není vztah nalezen, dojde k navrácení k původnímu dotazu a vyhledávání se opakuje obdobně pro matku první osoby. Pokud ani zde nebyl vztah nalezen, pokračuje vyhledávání stejně u druhé osoby.

Algoritmus tímto způsobem rekurzivně prohledává stavový prostor. Tato implementace je výpočetně poměrně dost náročná, vzhledem k tomu, že v každém kroku je znovu vyhledána otcovská či mateřská linie obou výchozích osob. Pro rozsáhlé genealogické databáze se tedy jedná o nepříliš efektivní algoritmus. Proto byl implementován jiný přístup.

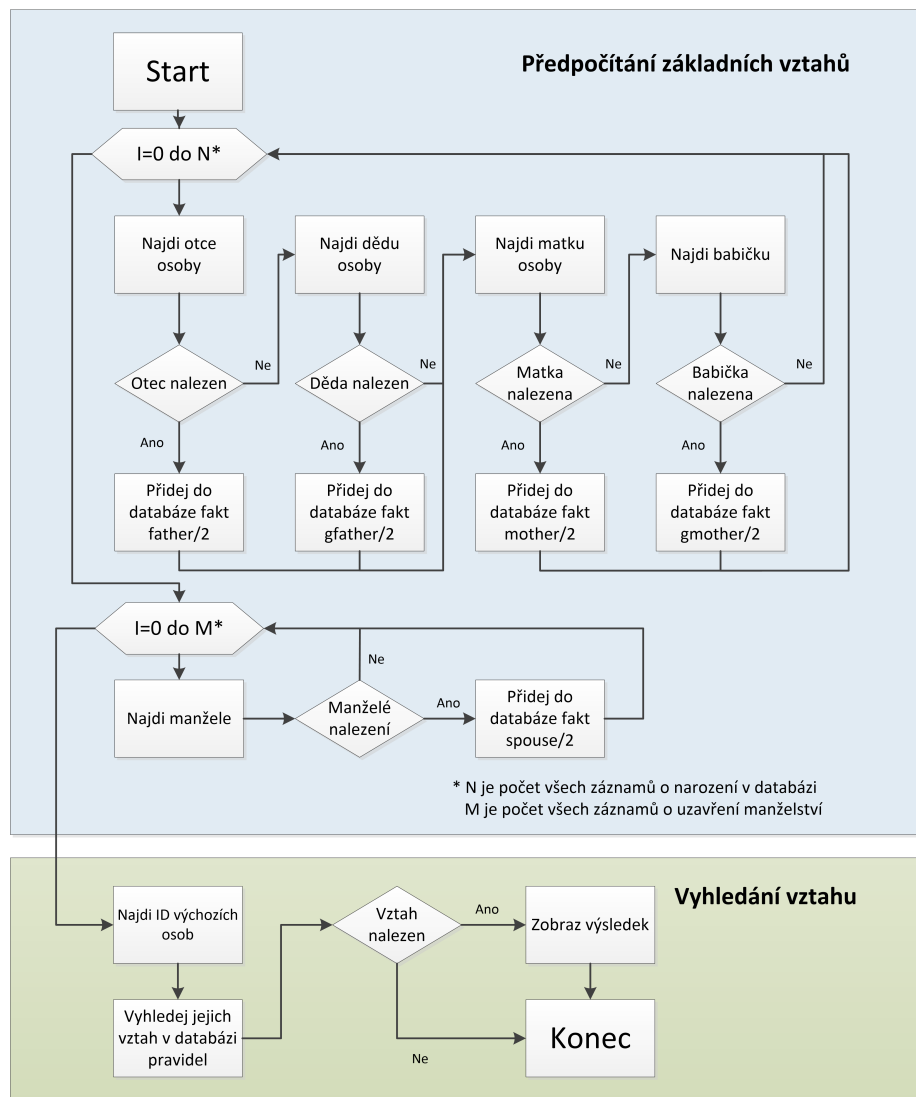
## Algoritmus č.2

Druhý algoritmus je implementován v dotazech `relation2/7` a `relation2/3`, kde opět jsou dvě osoby zadány buď celým jménem a rokem narození, anebo jejich identifikačním číslem.

Tento algoritmus se snaží minimalizovat výpočetní náročnost. Proto jsou základní vztahy všech osob v genealogické databázi předpočítány před samotným vyhledáváním. Základními vztahy rozumějme vyhledání otce a matky dané osoby. Tyto vztahy jsou následně zaneseny do databáze v podobě faktů `father/2` a `mother/2`. Stejně jako předchozí algoritmus, i tento se dokáže vyrovnat s chybějícím záznamem o narození otce či matky. Pokud nastane situace, že nebyl nalezen záznam o narození otce, je vyhledán děda. Stejně tak pokud chybí matka, je nalezena babička. Toto však ještě zcela neřeší problém a proto jsou v rámci předpočítání vztahů uloženy do databáze také fakta o manželských párech. Takový záznam je označen jako `spouse/2` a obsahuje identifikátory manželů. Díky tomu lze například dohledat vztah mezi výchozí osobou a její babičkou z otcovy strany i v případě, že databáze neobsahuje záznam o otcově narození. Je tedy vyhledán otcův otec a následně jeho manželka. Následně je pomocí těchto faktů implementována řada pravidel, která platí pro konkrétní vztahy, např. teta je sestra otce/matky, atd.

Výhodou tohoto algoritmu je především jeho jednoduchost. Vzhledem k tranzitivním vlastnostem v genealogických strukturách je postačující znalost rodičů každé osoby. Ostatní vztahy lze odvodit pomocí této znalosti. Vývojový diagram reprezentující tento algoritmus je na obrázku 4.15.

Algoritmus je navržen tak, aby uživatel mohl jednoduše definovat nové dotazy nad genealogickými daty. Stačí základní znalost jazyka Prolog a uživatel může dodefinovat pravidla, která platí pro další vztahy, podle jeho potřeb.



Obrázek 4.15: Vývojový diagram předpočítání základních vztahů.

## 4.7 Export databáze do souboru

Vzhledem k možnosti ručního vkládání záznamů bylo vhodné implementovat možnost exportu genealogické databáze do souboru. Veškeré dotazy zabezpečující export databáze jsou implementovány v souboru `export.pl`. Pro potřeby dalšího zpracování byly zvoleny dva formáty exportovaných souborů. Prvním je soubor s příponou `.pl`. Do souboru jsou data uložena ve stejné formě jako jsou reprezentována v interní databázi znalostí SWI-Prolog. Záznamy tedy odpovídají přesně struktuře popsané v kapitole 3.6. Takto exportovaný soubor lze později opět importovat a pracovat s ním. Pro export dat do tohoto formátu lze použít následující příkazy, které předpokládají jako parametr název výsledného souboru:

```

born_to_pl/1      - export databáze narozených
marriage_to_pl/1  - export databáze manželství
death_to_pl/1     - export databáze zemřelých

```

Druhým formátem pro exportování genealogických dat je soubor s příponou `.csv`. Jedná se o tzv. Comma-separated values soubor, kde je jako oddělovač použit středník. Soubor v tomto formátu lze otevřít v různých tabulkových editorech (např. Microsoft Excel, OpenOffice Calc, ...). Tento formát je tedy vhodný pro přenos genealogických dat a jejich další zpracování mimo vyvíjený systém. Pro export dat do tohoto formátu lze použít následující příkazy, které předpokládají jako parametr název výsledného souboru:

```
born_to_csv/1      - export databáze narozených
marriage_to_csv/1  - export databáze manželství
death_to_csv/1     - export databáze zemřelých
```

Lze rovněž provést export všech záznamů bez ohledu na jejich typ do jediného souboru. K tomu slouží příkaz `all_to_pl/1`. Varianta pro export do souboru `.csv` nebyla implementována, protože nemá smysl (nelze jedoduše rozlišit, který záznam je jakého typu a v tabulce nemá smysl mísit různé typy záznamů).

## 4.8 Manuál

Vzhledem ke skutečnosti, že nebylo implementováno grafické uživatelské rozhraní, není práce se systémem intuitivní. Aby mohl uživatel systém používat, musí mít představu jaké dotazy jsou implementovány a jak je správně používat. Kromě toho musí mít alespoň základní znalost jazyka Prolog a programování v něm. Právě z těchto důvodů byl implementován speciální dotaz `man/1`, který slouží jako rychlá nápověda. Jeho parametrem je název dotazu, pro který chce uživatel zobrazit nápovědu. Tento dotaz zobrazí příklad použití hledaného dotazu a všechny jeho implementované varianty s jejich krátkým popisem. U dotazů, u kterých to má smysl, je uveden formát či množina návratových hodnot. Příklad zobrazení nápovědy je na obrázku 4.16.

```
?- man(complete_check).

complete_check/0
  complete_check. -> consistency check of all records
complete_check/1
  complete_check(SURNAME). -> consistency check of all records with same surname
true.
```

Obrázek 4.16: Ukázka zobrazení nápovědy pro dotaz `complete_check`.

## 4.9 Další vývoj

Automatické zpracování genealogických dat je velmi širokým tématem, které nabízí mnoho různých oblastí dalšího rozvoje. Prvním cílem dalšího vývoje by jistě měla být optimalizace implementovaných algoritmů. V případě rozsáhlejších genealogických struktur by mohlo být jejich zpracování zdlouhavé a neefektivní.

Celý systém byl od počátku chápán jako logická vrstva rozsáhlejšího systému pro zpracování genealogických dat. Dalším krokem by tedy mělo být vytvoření vhodného uživatelského rozhraní, které by umožnilo jednodušší práci se systémem. Díky tomu, že SWI-Prolog implementuje rozsáhlé rozhraní pro C++, bylo by k tomuto účelu možno použít knihovnu Qt, která je určena pro vytváření programů s grafickým uživatelským rozhraním. Pro usnadnění integrace s grafickým uživatelským rozhraním byla u několika klíčových dotazů implementována možnost zadání osoby podle identifikačního čísla. [5] [6]

Dalším žádoucím rozšířením je možnost rekonstrukce složitějších genealogických struktur. To by bylo možné sloučením dílčích otcovských a mateřských linií, jejichž rekonstrukce již implementována je. Další možností rozšíření by bylo vyhledání rozrodu od konkrétní osoby (dle def. rozrodu v kap. 2.3). Ideálním výsledkem takového vývoje by byla automatická rekonstrukce genealogických struktur. Těžko v této chvíli říct, jestli je zcela automatická rekonstrukce možná, jistě však lze vytvořit systém, který by byl toho schopen s uživatelskou intervencí.

Dále by bylo rozhodně předmětem vývoje rozšíření sady pravidel pro určení vztahu mezi dvěma osobami. V rámci této práce byla implementována jen omezená množina pravidel, která nedokáže pojmenovat všechny možné vztahy v rodině. Implementovaná pravidla se zaměřují především na pokrevní příbuzenství, avšak bylo by vhodné implementovat pravidla pro právní příbuzenství označované jako švagrovství.

Z historického a genealogického hlediska by bylo jistě zajímavé rozšířit výpočet statistik o další zajímavé oblasti. Mohlo by se jednat o zastoupení jednotlivých zaměstnání či náboženská příslušnost na různých územích České republiky. Takových statistických údajů lze z genealogických záznamů zjistit celou řadu a vypovídají o životních podmínkách v dané době.

## Kapitola 5

# Experimenty nad výsledným systémem

V rámci testování byla nad výsledným systémem provedena řada experimentů. Jako vhodná testovací data posloužila jednoduchá genealogická struktura, která je zobrazena na obrázku 4.1. Díky jejímu malému rozsahu bylo možné jednoduše zkontrolovat správnost výstupů. Zároveň obsahuje dostatek informací pro řádné otestování funkčnosti systému. Následuje popis experimentů, které byly provedeny. U každého experimentu jsou porovnány očekávané výsledky s těmi reálnými.

### 5.1 Ověření konzistence dat

#### Experiment č.1 - Korektní data

První experiment se zaměřil na ověření funkčnosti systému nad korektními daty. Při spuštění kontroly nad korektními daty se očekává, že nebudou chybně detekovány chyby v konzistenci. Výsledek kompletní kontroly je na obrázku 5.1. Jak je vidět, systém správně vyhodnotil data jako konzistentní.

```
?- complete_check.  
OK: Check of IDs in born records  
OK: Check of IDs in marriage records  
OK: Check of IDs in deaths records  
  
OK: All fathers alive when descendants were born.  
OK: All mothers alive when descendants were born.  
OK: All mothers are of the correct age.  
OK: All fathers are of the correct age.  
OK: All siblings birth dates are valid.  
OK: Siblings in family have different names.  
OK: All marriages records are correct
```

Obrázek 5.1: Výsledek kontroly konzistence dat nad korektními daty

## Experiment č.2 - Nekonzistence mezi sourozenci

V tomto experimentu byla do databáze úmyslně zanesena chyba mezi sourozenci. Osoby David Prostředník a Andrej Prostředník jsou v databázi vedeny jako dvojčata. Jejich datum narození tedy musí být identické. Zanesenou chybou tedy bylo, že se jejich datum narození lišil o dva dny. Očekávanou reakcí by mělo být chybové hlášení. Systém tuto nekonzistenci vyhodnotil jako chybu a výsledné chybové hlášení je na obrázku 5.2. Stejné hlášení se objeví vždy, když rozestup mezi sourozenci bude menší než devět měsíců a více než třicet let.

```
Error: Siblings birth date not correct:
      26.5.2015 David Prostrednik (record 20)
      28.5.2015 Andrej Prostrednik (record 21)
```

Obrázek 5.2: Chybové hlášení nekonzistence v datu narození dvojčat.

Mezi sourozenci rovněž platí, že nesmí mít stejné křestní jméno. Proto bylo v databázi nastaveno stejné křestní jméno u těchto dvou osob. Očekávaným výsledkem je opět chybové hlášení. Systém chybu rozeznal a výsledek je zobrazen na obrázku 5.3.

```
Error: Siblings have same first name:
      28.5.2015 David Prostrednik (record 20)
      28.5.2015 David Prostrednik (record 21)
```

Obrázek 5.3: Chybové hlášení v případě stejného křestního jména sourozenců.

## Experiment č.3 - Nekonzistence mezi rodičem a potomkem

Dalším experimentem bylo ověření správnosti vztahu rodič-potomek. Nejdůležitější věcí, která musí platit v tomto vztahu, je správný věkový rozdíl mezi rodičem a potomkem. Proto byla do systému zanesena chyba, kdy se potomek narodil před narozením svého otce. Systém zareagoval chybovým hlášením, které je na obrázku 5.4.

```
Error: Father isn't the right age.
      Father: Roman Prostrednik born in 1987. (record 10)
      Descendant: David Prostrednik born in 1015. (record 20)
```

Obrázek 5.4: Chybové hlášení v případě špatného věkového rozestupu mezi rodičem a potomkem.

Stejně tak nesmí nastat situace, kdy se potomek narodil před narozením rodiče, případně, že se narodil po jeho smrti (u otce lze předpokládat, že se dítě narodí maximálně devět měsíců po jeho smrti). Dalším pokusem tedy bylo, jestli systém korektně odhalí chybu, v případě, že se potomek narodil až po smrti rodiče. Výsledek kontroly je na obrázku 5.5.

```
Error: Descendant born after mothers death:
      Mother death: Olga Sedlakova - 26.8.2005 (death record 2)
      Descendant born: Eva Sedlakova - 5.9.2015 (born record 3)
```

Obrázek 5.5: Chybové hlášení v případě narození potomka po smrti rodiče.

## Experiment č.4 - Nekonzistence mezi manželi

Poslední experiment v rámci kontroly konzistence má za úkol odhalit chyby v záznamech o uzavřených manželstvích. Platí, že manželé nesmí být sourozenci. Tato chyba se může lehce vyskytnout při špatném zápisu čísla rodiny (sourozenci mají stejné číslo rodiny). V rámci experimentu byla tato chyba zanesena do databáze. Stejně tak je chybou, pokud je některý z manželů v den svatby mladší patnácti let. Výsledek kontroly je na obrázku 5.6.

```
Error: Spouses are too young for marriage in 1959:
  Ivan Prostrednik born in 1957 (record 7)
  Eva Sedlakova born in 1958 (record 3)

Error: Spouses are probably siblings:
  Vladimir Sedlak born in 1933 (record 1)
  Olga Holeckova born in 1933 (record 2)
```

Obrázek 5.6: Chybové hlášení pokud jsou manželé sourozenci anebo jsou příliš mladí na uzavření manželství.

Další chybou může být špatně zapsané datum konání svatby. Chyba v tomto údaji potom může tvrdit, že se svatba stala po smrti některého z manželů, případně ještě před jejich narozením. I tato situace byla testována, výsledek zachycení takové chyby systémem je na obrázku 5.7.

```
Error: Marriage in 1930 happened before spouses were born.
  Ivan Prostrednik born in 1957 (record 7)
  Eva Sedlakova born in 1958 (record 3)

Error: Marriage in 2010 happened after spouses death.
  Olga Holeckova died in 2005 (death record 2)
```

Obrázek 5.7: Chybové hlášení pokud svatba nastala před narozením anebo po smrti některého z manželů.

## 5.2 Vyhledání otcovské/mateřské linie

### Experiment č.5 - Více vhodných kandidátů

Při vyhledávání otcovské či mateřské linie v rozsáhlejších databázích se může stát, že dojde k nalezení více vhodných osob, mezi kterými nelze jednoznačně rozhodnout, která je ta správná. V takovém případě se systém zachová takovým způsobem, že vyhledá od každé vhodné osoby linii dále. V rámci experimentu byla do databáze přidána další osoba Eva Sedláková. Potom v případě vyhledávání mateřské linie pro Hanu Prostředníkovou se vyskytnou dvě Evy Sedlákové, které obě vyhovují tomu aby byly zvoleny matkou. Na obrázku 5.8 je zobrazen výsledek vyhledání takové ženské linie.

```
?- get_femine_line(11,L).
[[['Hana', 'Prostrednikova', 1993, 7, 11], [['Eva', 'Sedlakova', 1958, 3, 3], [['Olga', 'Holeckova', 1933, 2, 2]]],
 [['Eva', 'Sedlakova', 1959, 6, 26]]]
```

Obrázek 5.8: Výsledek vyhledání mateřské linie s více vhodnými osobami.



### 5.3 Určení vztahu mezi dvěma osobami

#### Experiment č.6 - Určení vztahu pomocí kompletních dat

Při experimentování s vyhledáváním vztahu mezi dvěma osobami byla nejdříve použita kompletní a korektní data. To znamená, že při vyhledávání otcovské či mateřské linie jsou přítomny záznamy o všech osobách. V kapitole 4.6 byly uvedeny dvě varianty algoritmu pro vyhledávání vztahu mezi osobami. Nutno podotknout, že od algoritmu č. 1 bylo při vývoji upuštěno pro jeho nízkou efektivitu a zbytečnou složitost. Proto bylo implementováno pouze vyhledání několika základních vztahů. Pro úplnost však byly i nad ním provedeny experimenty. Mezi osobami Hana Prostředníková (ID 11) a Eva Sedláková (ID 3) je podle obrázku 4.1 vztah rodič-potomek. Mezi osobami Hana Prostředníková a Jitka Sedláková (ID 4) je potom vztah tetá-neteř. Výsledek vyhledání těchto vztahů odpovídá očekáváním a je zobrazen na obrázku 5.9.

```
?- relation(11,3,L).
Hana Prostrednikova born in 1993 and Eva Sedlakova born in 1958
  Relation: parent - descendant
L = 1.

?- relation(11,4,L).
Hana Prostrednikova born in 1993 and Jitka Sedlakova born in 1967
  Relation: aunt/uncle - niece/nephew
L = 102.
```

Obrázek 5.9: Výsledek vyhledání vztahu mezi osobami pomocí algoritmu č.1 z kapitoly 4.6.

Dále byli provedeny experimenty nad algoritmem č.2. Osoby Hana Prostředníková (ID 11) a Daniel Pernet (ID 12) jsou dle obrázku 4.1 ve vztahu bratranec-sestřenice. Systém tento vztah vyhledal správně (obr 5.10 první dotaz). Dále stejně jako bylo uvedeno výše, osoby s identifikačními čísly 11 a 3 jsou ve vztahu matka-potomek. Rovněž tento vztah systém vyhledal správně. Jak je vidět na obrázku 5.10 v druhém dotazu, výhodou tohoto algoritmu je, že vyhledá všechny vztahy, ve kterých osoby jsou. Posledním dotazem je vztah mezi osobou Hana Prostředníková (ID 11) a František Sedlák (ID 14). Mezi osobami je vztah praděda-pravnučka. I tento vztah systém vyhledal správně, jak je vidět na obrázku 5.10.

```
?- relation2(11,12,L).
L = cousins .

?- relation2(11,3,L).
L = 'parent-descendant' ;
L = 'mother-descendant' ;
true.

?- relation2(11,14,L).
L = 'great_grandparent-great_grandchild' ;
true.
```

Obrázek 5.10: Výsledek vyhledání vztahu mezi osobami pomocí algoritmu č.2 z kapitoly 4.6.

## Experiment č.7 - Určení vztahu pomocí nekompletních dat

Vzhledem k schopnosti systému vyrovnat se do jisté míry s chybějícím záznamem o narození některého z předků v otcovské či mateřské linii, je vhodné experimentovat s touto vlastností. Při pokusech byly z databáze odstraněny záznamy o narození osob Ivan Prostředník (ID 7) a Olga Holečková (ID 2). Záznamy o uzavření sňatků byly v databázi ponechány.

Jako první byl opět otestován první algoritmus z kapitoly 4.6. Jak je vidět na obrázku 5.11 výsledek prvního dotazu je správný i když chybí záznam o narození osoby Ivan Prostředník, který je otcem Hany a synem Jana. Podobně je správně vyhledán vztah mezi osobami Hana Prostředníková a Jitka Sedláková. Tyto dvě osoby mají společnou část mateřské linie, do které patří i Olga Holečková (matka Jitky a babička Hany), jejíž záznam o narození však chybí. Stejný experiment byl proveden u druhého implementovaného algoritmu. Jak je vidět na obrázku 5.12, i v tomto případě bylo vyhledání vztahů úspěšné a správné.

```
?- relation(11,6,L).
Hana Prostrednikova born in 1993 and Jan Prostrednik born in 1938
  Relation: grandparent - grandchild
L = 2.

?- relation(11,4,L).
Hana Prostrednikova born in 1993 and Jitka Sedlakova born in 1967
  Relation: aunt/uncle - niece/nephew
L = 102.
```

Obrázek 5.11: Výsledek vyhledání vztahu mezi osobami pomocí algoritmu č.1 z kapitoly 4.6.

```
?- relation2(11,1,L).
L = 'grandparent-grandchild' ;
true.

?- relation2(11,4,L).
L = aunt_uncle ;
true.
```

Obrázek 5.12: Výsledek vyhledání vztahu mezi osobami pomocí algoritmu č.2 z kapitoly 4.6.

Pro úplnost bylo vhodné provést experiment, kdy chybí dvě osoby v otcovské či mateřské linii za sebou. Proto byl z databáze odstraněn jak záznam o narození Ivana Prostředníka, tak i záznam o narození Romana Prostředníka. Následně byl vyhledán vztah mezi Davidem a Janem Prostředníkem. Tyto dvě osoby jsou ve vztahu praděd-pravnuk. Z obrázku 5.13 je patrné, že ani jeden algoritmus nebyl úspěšný při vyhledání. Důvodem je, že oba algoritmy používají vyhledávání otcovské linie. Ta však nemůže být správně vyhledána, pokud chybí dvě a více osob za sebou. K vyhledání praoťce bez znalosti otce a dědy neexistuje dostatek informací.

```

?- relation(20,6,L).
David Prostrednik born in 2015 and Jan Prostrednik born in 1938
  Relation: not found
true.

?- relation2(20,6,L).
true.

```

Obrázek 5.13: Výsledek vyhledání vztahu mezi osobami pomocí algoritmu č.2 z kapitoly 4.6.

## 5.4 Statistiky

### Experiment č.8

Jak bylo zmíněno v kapitole 4.4, lze zobrazovat statistiky dle konkrétního příjmení. Pokud bychom tedy zobrazili statistiku o narozených osobách dle příjmení Prostředník, měla by statistika brát v úvahu osm osob (dle 4.1). Výsledek této statistiky je na obrázku 5.14.

```

?- born_statistics('Prostrednik').

Statistics of births

Total born records count: 8
Recorded years: 1860 - 2015
The most borns in 2015. Count: 2
The least borns in 1938. Count: 1
Average borns count per year: 0.05161290322580645
The most frequent name is Andrej. Count: 1
true.

```

Obrázek 5.14: Statistika o narozených osobách dle příjmení Prostředník.

Statistika na malém objemu ztrácí své kouzlo a proto byl dotaz otestován na rozsáhlejší databázi. Tato databáze obsahuje záznamy o narození v průběhu přibližně sta let. Výsledek statistiky je na obrázku 5.15. Stejně tak byla zobrazena statistika i o uzavřených manželstvích (obr. 5.16). Bohužel databáze obsahující větší objem záznamů o zemřelých není k dispozici.

```

?- born_statistics.

Statistics of births

Total born records count: 4389
Recorded years: 1686 - 1784
The most borns in 1696. Count: 65
The least borns in 1730. Count: 5
Average borns count per year: 44.785714285714285
The most frequent name is WOLF. Count: 168
The most frequent name is Maria. Count: 316
true.

```

Obrázek 5.15: Statistika o narozených osobách na větším objemu dat.

```
?- marriage_statistics.  
  
Statistics of marriages  
  
Total marriage records count: 1169  
Recorded years: 1687 - 1784  
The most marriages in 1691. Count: 21  
The least marriages in 1712. Count: 3  
Average marriages count per year: 12.051546391752577  
The most descendants has couple: Fabian Schallamon and Maria Lerch. Count: 14  
Average descendant count per pair: 4.053030303030303  
true.
```

Obrázek 5.16: Statistika o uzavřených manželstvích na větším objemu dat.

## Kapitola 6

### Závěr

Cílem práce bylo studium problematiky genealogie a způsobu reprezentace genealogických dat. Byly definovány základní pojmy a uvedeny všechny potřebné informace. Bylo potřebné nastudovat nezbytné informace týkající se jak genealogie, tak i jazyka Prolog a práce s ním. Všechny tyto informace byly použity při implementaci výsledného systému.

V průběhu návrhu systému byl kladen důraz na správnou specifikaci vstupních dat. Struktura dat byla odvozena od dat, která poskytl vedoucí této práce doc. Zbořil. Nicméně tato struktura je velmi závislá na období, ze kterého data pocházejí, a proto není vyloučeno, že při dalším vývoji bude změněna. Důležité bylo nalezení a definice pravidel platných v genealogických strukturách.

Výsledkem je funkční systém, který odpovídá všem bodům zadání. Uživatel má možnost importovat vlastní genealogickou databázi. V této databázi následně může kontrolovat platnost pravidel jednotlivých vztahů a docílit tak, aby byla data v databázi konzistentní. Kromě toho lze systém použít pro rekonstrukci některých jednoduchých genealogických struktur a pro vyhledání vztahu mezi dvěma osobami. Byly implementovány různé přístupy, jejichž funkčnost byla diskutována.

Další vývoj by měl směřovat k rozsáhlejší kontrolám konzistence dat. Dále by měla být předmětem vývoje co nejpřesnější automatická rekonstrukce genealogických struktur. Vhodná by byla také optimalizace implementovaných algoritmů a celkové rozšíření systému pro jeho další použití ve větším genealogickém systému.

Velkým osobním přínosem pro mne bylo důkladnější studium jazyka Prolog, který podle mého názoru poskytuje mnoho zajímavých možností pro vývoj aplikace tohoto charakteru.

# Literatura

- [1] *Holocaust: Norimberské zákony (1935)* . [Online; navštíveno 20.12.2017].  
URL <http://www.genea.cz/informace/badani-v-archivu/matriky/>
- [2] *Matriky. Genea.* [Online; navštíveno 18.12.2017].  
URL <http://www.holocaust.cz/dejiny/udalosti/norimberske-zakony-1935/>
- [3] *Moderní dějiny: Doklad árijského původu - Arier-Nachweis (1941).* [Online; navštíveno 20.12.2017].  
URL <http://www.moderni-dejiny.cz/clanek/doklad-arijskeho-puvodu-arier-nachweis-1941/>
- [4] *Nejstarší matrika. Po stopách předků.* [Online; navštíveno 20.12.2017].  
URL <http://www.postopachpredku.cz/?p=362>
- [5] *SWI-Prolog's features.* [Online; navštíveno 5.4.2018].  
URL <http://www.swi-prolog.org/features.html>
- [6] *The Qt Company - Ensuring the future is written with Qt* . [Online; navštíveno 25.04.2018].  
URL <https://www1.qt.io/company/>
- [7] *The Gedcom Standard.* 1989, [Online; navštíveno 05.02.2018].  
URL <https://chronoplexsoftware.com/gedcomvalidator/gedcom/gedcom-4.0.pdf>
- [8] *The Gedcom Specification.* The Church of Jesus Christ of Latter-day Saints, 2000, [Online; navštíveno 02.04.2016].  
URL <https://chronoplexsoftware.com/gedcomvalidator/gedcom/gedcom-5.6.pdf>
- [9] *Prolog. LinuxEXPRES* . 2006, [Online; navštíveno 21.12.2017].  
URL <https://www.linuxexpres.cz/praxe/prolog>
- [10] *Státní oblastní archiv v Praze.* 2014, [Online; navštíveno 25.01.2018].  
URL <http://ebadatelna.soapraha.cz>
- [11] Czensky, A.: *Prolog Techniques*. Ventus Publishing ApS, 2009, ISBN 978-87-7681-476-2.
- [12] Juraj Snopek, M. T.: *Ako si zostaviť rodokmeň: Spoznajete históriu svojho rodu.* Bratislava: IKAR, 2017, ISBN 978-80-551-5749-8.
- [13] Kolář, D.: *Funkcionální a logické programování: Studijní opora* . Fakulta informačních technologií, VUT v Brně.

URL <https://wis.fit.vutbr.cz/FIT/st/course-files-st.php?file=%2Fcourse%2FFLP-IT%2Ftexts%2FFPR-Logicke.pdf>.

- [14] Kovár, M.: *Diskrétní matematika: Studijní opora* . Fakulta informačních technologií, VUT v Brně.  
URL <http://www.umat.feec.vutbr.cz/~kovar/webs/personal/IDA.pdf>
- [15] Kvasil, B.; aj.: *Malá československá encyklopedie*. Academia, 1985, ISBN 21-118-84.
- [16] Roháček, J.: *Svätá Biblia*. Svetová biblická spoločnosť, 1969, ISBN 978-80-8548-686-5, revidované vydanie.
- [17] Sperát, I.: *Rokodmeny - kompletní genealogické služby*. 2018, [Online; navštíveno 26.3.2018].  
URL <http://www.sperat.cz/rodokmeny/>
- [18] Tišer, J.: *Základní pojmy z teorie grafů* . [Online; navštíveno 20.12.2017].  
URL <http://math.feld.cvut.cz/tiser/Grafy.pdf>
- [19] Český statistický úřad: *Zaostřeno na ženy a muže - Živě narození, potraty a ukončená těhotenství podle věku žen*. 2016, [Online; navštíveno 11.05.2018].  
URL <https://www.czso.cz/documents/10180/32853391/300002161114.pdf/2175181a-eecc-4748-b107-c9370957a8a2?version=1.1>

# Přílohy



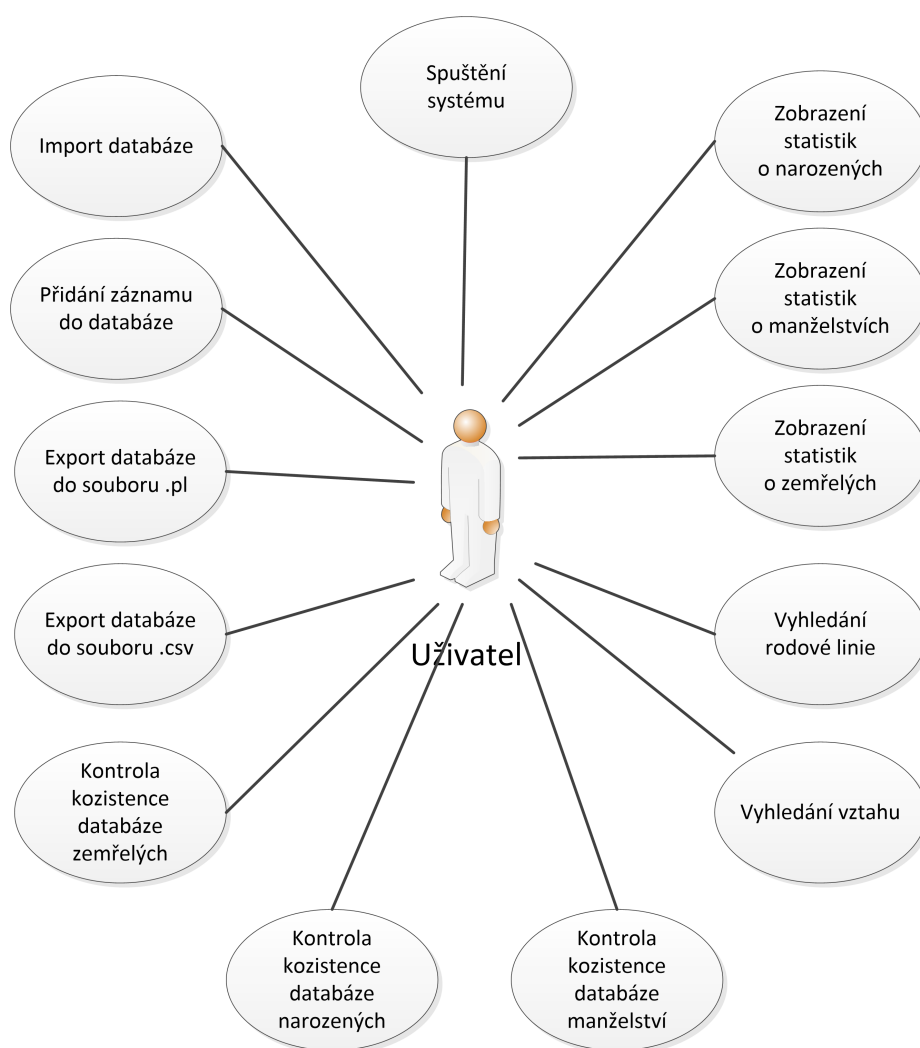
## Příloha A

# Obsah přiloženého CD

- /src - zdrojové kódy včetně testovacích dat
- /text\_pdf - technická zpráva ve formátu pdf
- /text\_src - zdrojový tvar technické zprávy
- /poster - plakát
- test\_data.png - obrázek struktury testovacích dat
- readme.txt - návod k instalaci

## Příloha B

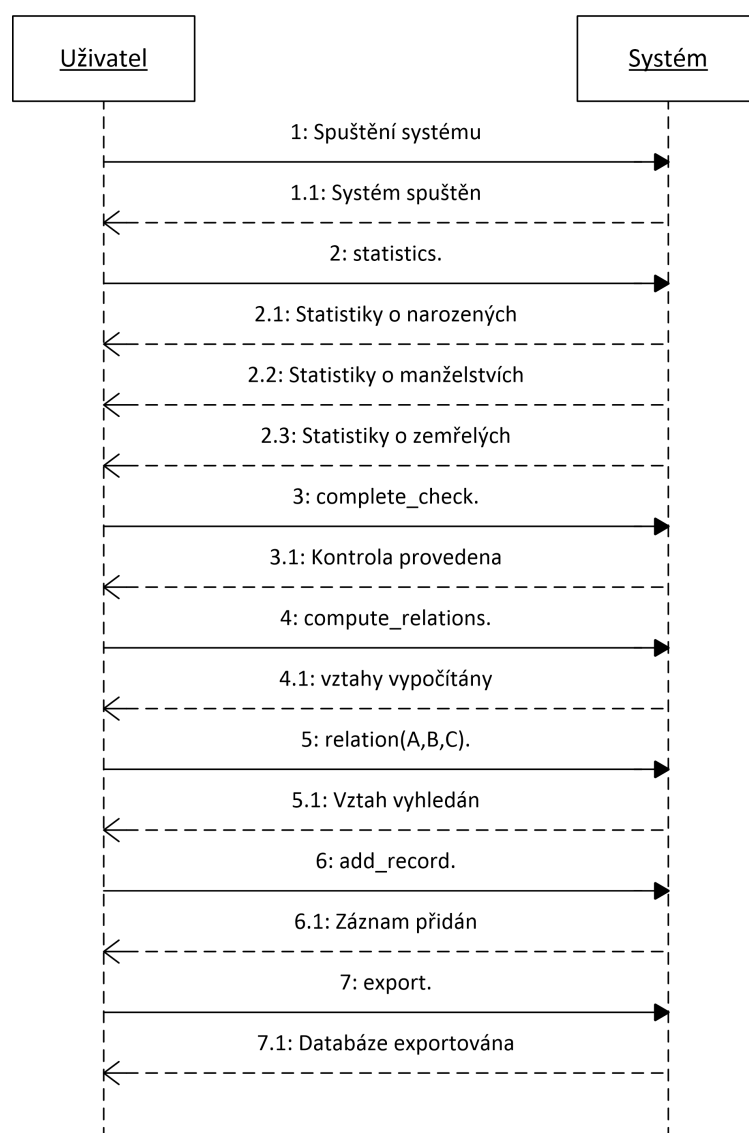
### Use-case diagram



Obrázek B.1: Use case diagram

## Příloha C

# Systémový sekvenční diagram



Obrázek C.1: Systémový sekvenční diagram